ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

CSRN 2602

(Ed.)

- **Vaclav Skala**
  University of West Bohemia, Czech Republic

Computer Science Research Notes

24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2016
Plzen, Czech Republic
May 30 – June 3, 2016

Proceedings

WSCG 2016

Short Papers Proceedings

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

CSRN 2602

(Ed.)

- **Vaclav Skala**
  University of West Bohemia, Czech Republic

Computer Science Research Notes

24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2016
Plzen, Czech Republic
May 30 – June 3, 2016

Proceedings

WSCG 2016

Short Papers Proceedings

Vaclav Skala – Union Agency

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

Computer Science Research Notes
CSRN 2602

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz *http://www.VaclavSkala.eu*

Managing Editor: Vaclav Skala

Publisher & Author Service Department & Distribution:
Vaclav Skala - UNION Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic
Reg.No. (ICO) 416 82 459

ISSN 2464-4617 (Print) **ISSN 2464-4625** (CD/DVD)
ISBN 2464-4625 (CD ROM**)** **ISBN 978-80-86943-57-2** (CD/-ROM)

WSCG 2016
International Program Committee

Benger,Werner (United States)
Bilbao,Javier,J. (Spain)
Bittner,Jiri (Czech Republic)
Buehler,Katja (Austria)
Chaudhuri,Debasis (India)
Chmielewski,Leszek (Poland)
Coquillart,Sabine (France)
Daniel,Marc (France)
de Geus,Klaus (Brazil)
Dingliana,John (Ireland)
Drechsler,Klaus (Germany)
Durikovic,Roman (Slovakia)
Falcidieno,Bianca (Italy)
Feito,Francisco (Spain)
Ferguson,Stuart (United Kingdom)
Flaquer,Juan (Spain)
Garcia Hernandez,Ruben Jesus (Germany)
Giannini,Franca (Italy)
Gudukbay,Ugur (Turkey)
Guthe,Michael (Germany)
Jeschke,Stefan (Austria)
Juan,M.-Carmen (Spain)
Kim,H. (Korea)
Max,Nelson (United States)
Molla,Ramon (Spain)
Montrucchio,Bartolomeo (Italy)
Muller,Heinrich (Germany)
Murtagh,Fionn (United Kingdom)
Pan,Rongjiang (China)
Pedrini,Helio (Brazil)
Platis,Nikos (Greece)
Renaud,christophe (France)
Richardson,John (United States)
Ritter,Marcel (Austria)
Rojas-Sola,Jose Ignacio (Spain)
Sanna,Andrea (Italy)
Segura,Rafael (Spain)
Skala,Vaclav (Czech Republic)
Slavik,Pavel (Czech Republic)
Sousa,A.Augusto (Portugal)
Szecsi,Laszlo (Hungary)
Teschner,Matthias (Germany)
Tokuta,Alade (United States)
Trapp,Matthias (Germany)
Viola,Ivan (Austria)
Wu,Shin-Ting (Brazil)
Wuensche,Burkhard,C. (New Zealand)
Wuethrich,Charles (Germany)
Zara,Jiri (Czech Republic)
Zemcik,Pavel (Czech Republic)

WSCG 2016
Board of Reviewers

Abad,Francisco (Spain)
Adzhiev,Valery (United Kingdom)
Agathos,Alexander (Romania)
Ahmad,Khurshid (Ireland)
Akleman,Ergun (United States)
Amditis,Angelos (Greece)
Ammi,Mehdi (France)
Ammi,Mehdi (France)
Ariu,Davide (Italy)
Assarsson,Ulf (Sweden)
Aveneau,Lilian (France)
Ayala,Dolors (Spain)
Backfrieder,Werner (Austria)
Barthe,Loic (France)
Battiato,Sebastiano (Italy)
Baum,David (Germany)
Benes,Bedrich (United States)
Benger,Werner (United States)
Benoit,Crespin (France)
Biasotti,Silvia (Italy)
Bilbao,Javier,J. (Spain)
Biri,Venceslas (France)
Birra,Fernando (Portugal)
Bittner,Jiri (Czech Republic)
Bosch,Carles (Spain)
Bouatouch,Kadi (France)
Boukaz,Saida (France)
Bourdin,Jean-Jacques (France)
Bourke,Paul (Australia)
Bouville,Christian (France)
Bruckner,Stefan (Austria)
Bruder,Gerd (Germany)
Brun,Anders (Sweden)
Bruni,Vittoria (Italy)
Brunnett,Guido (Germany)
Buehler,Katja (Austria)
Bulo,Samuel Rota (Italy)
Buriol,Tiago Martinuzzi (Brazil)
Cakmak,Hueseyin (Germany)
Camahort,Emilio (Spain)
Casciola,Giulio (Italy)
Chaine,Raphaelle (France)
Chaudhuri,Debasis (India)
Chen,Falai (China)
Chmielewski,Leszek (Poland)
Choi,Sunghee (Korea)
Chover,Miguel (Spain)
Chrysanthou,Yiorgos (Cyprus)
Chuang,Yung-Yu (Taiwan)
Cline,David (United States)
Coquillart,Sabine (France)
Corcoran,Andrew (Ireland)
Cosker,Darren (United Kingdom)
Daniel,Marc (France)
Daniels,Karen (United States)
de Amicis,raffaele (Italy)
de Geus,Klaus (Brazil)
de Oliveira Neto,Manuel Menezes (Brazil)
De Paolis,Lucio Tommaso (Italy)
Debelov,Victor (Russia)
Dingliana,John (Ireland)
Doellner,Juergen (Germany)
Dokken,Tor (Norway)
Drechsler,Klaus (Germany)
Durikovic,Roman (Slovakia)
Eisemann,Martin (Germany)
Erleben,Kenny (Denmark)
Falcidieno,Bianca (Italy)
Faudot,Dominique (France)
Feito,Francisco (Spain)
Ferguson,Stuart (United Kingdom)
Fiorentino,Michele (Italy)
Flaquer,Juan (Spain)
Fuenfzig,Christoph (Germany)
Gain,James (South Africa)
Galo,Mauricio (Brazil)
Garcia Hernandez,Ruben Jesus (Germany)
Garcia-Alonso,Alejandro (Spain)
Gavrilova,M. (Canada)
Gianelli,Carlota (Germany)
Giannini,Franca (Italy)

Gobron,Stephane (Switzerland)
Goebel,Martin (Germany)
Gonzalez,Pascual (Spain)
Grau,Sergi (Spain)
Gu,Xianfeng (United States)
GuĂŠrin,Eric (France)
Gudukbay,Ugur (Turkey)
Guthe,Michael (Germany)
Habel,Ralf (Switzerland)
Hall,Peter (United Kingdom)
Hansford,Dianne (United States)
Haro,Antonio (United States)
Hast,Anders (Sweden)
Hauser,Helwig (Norway)
Havemann,Sven (Austria)
Havran,Vlastimil (Czech Republic)
Hege,Hans-Christian (Germany)
Hernandez,Benjamin (United States)
Herout,Adam (Czech Republic)
Hicks,Yulia (United Kingdom)
Hildenbrand,Dietmar (Germany)
Hinkenjann,Andre (Germany)
Horain,Patrick (France)
Horain,Patrick (France)
House,Donald (United States)
Ihrke,Ivo (Germany)
Iwasaki,Kei (Japan)
Jeschke,Stefan (Austria)
Jiang,Jianmin (China)
Jones,Mark (United Kingdom)
Juan,M.-Carmen (Spain)
Juettler,Bert (Austria)
Kanai,Takashi (Japan)
Kim,H. (Korea)
Klosowski,James (United States)
Kohout,Josef (Czech Republic)
Kolcun,Alexej (Czech Republic)
Krueger,Jens (Germany)
Kumar,Subodh (India)
Kurillo,Gregorij (United States)
Kurt,Murat (Turkey)
Kyratzi,Sofia (Greece)
Lanquentin,Sandrine (France)
Larboulette,Caroline (France)
Lee,Jong Kwan Jake (United States)
Lengyel,Eric (United States)
Lien,Jyh-Ming (United States)
Lindow,Norbert (Germany)
Liu,SG (China)
Liu,Damon Shing-Min (Taiwan)
Lopes,Adriano (Portugal)
Loscos,Celine (France)
Lucas,Laurent (France)
Lutteroth,Christof (New Zealand)
Maciel,Anderson (Brazil)
Maddock,Steve (United Kingdom)
Magnor,Marcus (Germany)
Manak,Martin (Czech Republic)
Mandl,Thomas (Germany)
Manzke,Michael (Ireland)
Mas,Albert (Spain)
Masia,Belen (Spain)
Masood,Syed Zain (United States)
Matey,Luis (Spain)
Matkovic,Kresimir (Austria)
Max,Nelson (United States)
McDonnell,Rachel (Ireland)
McKisic,Kyle (United States)
Meng,Weiliang (China)
Mestre,Daniel,R. (France)
Metodiev,Nikolay Metodiev (United States)
Meyer,Alexandre (France)
Mokhtari,Marielle (Canada)
Molina Masso,Jose Pascual (Spain)
Molla,Ramon (Spain)
Montrucchio,Bartolomeo (Italy)
Morigi,Serena (Italy)
Muller,Heinrich (Germany)
Murtagh,Fionn (United Kingdom)
Myszkowski,Karol (Germany)
Niemann,Henrich (Germany)
Nishita,Tomoyuki (Japan)
Okabe,Makoto (Japan)
Oliveira, Jr.,Pedro Paulo (Brazil)
Oyarzun Laura,Cristina (Germany)
Pala,Pietro (Italy)
Pan,Rongjiang (China)
Papaioannou,Georgios (Greece)
Paquette,Eric (Canada)
Pasko,Alexander (United Kingdom)
Pasko,Galina (United Kingdom)

Pastor,Luis (Spain)
Patane,Giuseppe (Italy)
Patow,Gustavo (Spain)
Pedrini,Helio (Brazil)
Pereira,Joao Madeiras (Portugal)
Perret,Jerome (France)
Peters,Jorg (United States)
Pettre,Julien (France)
Peytavie,Adrien (France)
Pina,Jose Luis (Spain)
Platis,Nikos (Greece)
Plemenos,Dimitri (France)
Post,Frits,H. (Netherlands)
Poulin,Pierre (Canada)
Praktikakis,Ioannis (Greece)
Puig,Anna (Spain)
Puppo,Enrico (Italy)
Rafferty,Karen (United Kingdom)
Reisner-Kollmann,Irene (Austria)
Renaud,christophe (France)
Renaud,christophe (France)
Reyes-Lecuona,Arcadio (Spain)
Richardson,John (United States)
Ritschel,Tobias (Germany)
Ritter,Marcel (Austria)
Rojas-Sola,Jose Ignacio (Spain)
Rokita,Przemyslaw (Poland)
Runde,Christoph (Germany)
Ruther,Heinz (South Africa)
Sacco,Marco (Italy)
Sadlo,Filip (Germany)
Sakas,Georgios (Germany)
Salvetti,Ovidio (Italy)
Sanna,Andrea (Italy)
Santos,Luis Paulo (Portugal)
Sapidis,Nickolas,S. (Greece)
Savchenko,Vladimir (Japan)
Schultz,Thomas (Germany)
Schumann,Heidrun (Germany)
Segura,Rafael (Spain)
Seipel,Stefan (Sweden)
Sellent,Anita (Switzerland)
Shesh,Amit (United States)
Sik-Lanyi,Cecilia (Hungary)
Slavik,Pavel (Czech Republic)
Sochor,Jiri (Czech Republic)
Solis,Ana Luisa (Mexico)
Sommer,Bj?rn (Germany)
Sourin,Alexei (Singapore)
Sousa,A.Augusto (Portugal)
Sramek,Milos (Austria)
Sreng,Jean (France)
Staadt,Oliver (Germany)
Stricker,Didier (Germany)
Stroud,Ian (Switzerland)
Subsol,Gerard (France)
Suescun,Angel ()
Sunar,Mohd-Shahrizal (Malaysia)
Sundstedt,Veronica (Sweden)
Svoboda,Tomas (Czech Republic)
Szecsi,Laszlo (Hungary)
Tang,Min (China)
Tang,Qian (China)
Taubin,Gabriel (United States)
Tavares,Joao Manuel R.S. (Portugal)
Teschner,Matthias (Germany)
Theussl,Thomas (Saudi Arabia)
Tian,Feng (United Kingdom)
Tobler,Robert (Austria)
Todt,Eduardo (Brazil)
Tokuta,Alade (United States)
Torrens,Francisco (Spain)
Trapp,Matthias (Germany)
Triantafyllidis,Georgios (Greece)
Tytkowski,Krzysztof (Poland)
Umlauf,Georg (Germany)
Vanderhaeghe,David (France)
Vasa,Libor (Czech Republic)
Vazquez,Pere-Pau (Spain)
Vazquez,Pere Pau (United States)
Viola,Ivan (Austria)
Vitulano,Domenico (Italy)
Vosinakis,Spyros (Greece)
Walczak,Krzysztof (Poland)
WAN,Liang (China)
Wang,Charlie,C.L. (Hong Kong SAR)
Weber,Andreas (Germany)
Wenger,Raphael (United States)
Westermann,Ruediger (Germany)
Wu,Enhua (China)
Wu,Shin-Ting (Brazil)
Wuensche,Burkhard,C. (New Zealand)

Wuethrich,Charles (Germany)
Xin,Shi-Qing (Singapore)
Yoshizawa,Shin (Japan)
YU,Qizhi (United Kingdom)
Yue,Yonghao (Japan)
Zachmann,Gabriel (Germany)
Zalik,Borut (Slovenia)
Zara,Jiri (Czech Republic)
Zemcik,Pavel (Czech Republic)
Zhang,Xiaopeng (China)
Zhang,Xinyu (United States)
Zhu,Ying (United States)
Zillich,Michael (Austria)
Zitova,Barbara (Czech Republic)
Zwettler,Gerald (Austria)

WSCG 2016

Short Papers Proceedings

Contents

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Polynomiography for Square Systems of Equations with Mann and Ishikawa Iterations

Krzysztof Gdawiec
Institute of Computer Science
University of Silesia
Bedzinska 39
41-200, Sosnowiec, Poland
kgdawiec@ux2.math.us.edu.pl

Wiesław Kotarski
Institute of Computer Science
University of Silesia
Bedzinska 39
41-200, Sosnowiec, Poland
kotarski@ux2.math.us.edu.pl

Agnieszka Lisowska
Institute of Computer Science
University of Silesia
Bedzinska 39
41-200, Sosnowiec, Poland
alisow@ux2.math.us.edu.pl

ABSTRACT

In this paper we propose to replace the standard Picard iteration in the Newton–Raphson method by Mann and Ishikawa iterations. This iteration's replacement influence the solution finding process that can be visualized as polynomiographs for the square systems of equations. Polynomiographs presented in the paper, in some sense, are generalization of Kalantari's polynomiography from a single polynomial equation to the square systems of equations. They are coloured based on two colouring methods: basins of attractions with different colours for every real root and colouring dependent on the number of iterations. Possible application of the presented method can be addressed to computer graphics where aesthetic patterns can be used in e.g. texture generation, animations, tapestry design.

Keywords

Mann iteration, Ishikawa iteration, polynomiography, computer graphics

1 INTRODUCTION

Kalantari [Kal05a, Kal08] defined polynomiography as the art and science of visualization in approximation of the zeros of complex polynomials via fractal and non–fractal images created using mathematical convergence properties of iteration functions. As iteration functions the well–known Newton method, methods from Basic Family and Euler–Schröder Family of Iterations can be used. The polynomiograph is a single image that presents visualization process of roots finding for some polynomial. Polynomiographs are two–dimensional images generated in complex plane. Polynomiography, as a method producing nicely looking graphics was patented by Kalantari in the USA in 2005 [Kal05a]. Moreover, it found applications in: creating paintings, carpet design, tapestry design, animations etc. [Kal05b].

In [GKL14, GKL15] the authors presented a survey of some modifications of Kalantari's polynomiography based on the classic Newton's and the higher order Newton-like root finding methods for complex polynomials. Instead of the standard Picard's iteration several different iteration processes were used. By combining different kinds of iterations, different convergence tests, and different colouring they obtained a great variety of polynomiographs [GKL15].

In this paper we discuss a possibility of using the root finding visualization process for square systems of equations with two variables. For such a system the Newton–Raphson method [CK08, SF10] with standard Picard iteration is applicable and works well. In the proposed modification we replaced Picard iteration by Mann and Ishikawa ones. We do not investigate properties of numerical methods after the change of iterations. Mann and Ishikawa iterations are used to perturb the shape of polynomiographs and make them to look more interesting and aesthetically pleasing. So, the main aim of the paper is only to create artistic images.

Usually, for the Newton–Raphson method several iterations are needed to obtain a good accuracy of roots finding approximations. It should be mentioned that the Newton–Raphson method is applicable to more general case – to square systems of equations with any finite number of variables. But visualizations will be presented only in two dimensions, in the plane with two real axes. So, in the sequel, only systems of two equations with two variables are taken into account. This limitation is the first drawback of the method. The second one is the fact that polynomiographs for square systems of equations with two variables can visualize only the real roots of the square systems.

It should be pointed out that the full control of polynomiograph is possible only for the case if the square system has only real roots. Such a situation occurs e.g. for the system:

$$\begin{cases} x(y-1)(x-1)=0, \\ y(y+1)(x+1)=0 \end{cases} \tag{1}$$

having the following five real solutions:

$$\{0,0\},\{0,-1\},\{1,0\},\{1,-1\},\{-1,1\}.$$

The paper is organized as follows. In Sec. 2 Mann and Ishikawa iterations are defined. Sec. 3 presents formulas for Newton–Raphson method and their generalizations obtained using Mann and Ishikawa iterations instead of the standard Picard iteration. In Sec. 4 some examples of polynomiographs are presented. Sec. 5 concludes the paper and shows the future research directions.

2 ITERATIONS

Let $w : X \to X$ be a mapping on a metric space (X,d), where d is a metric. Further, let $u_0 \in X$ be a starting point. Following [Ber07] we recall some popular iterative procedures.

- Picard iteration:

$$u_{n+1} = w(u_n), \quad n = 0,1,2,\ldots, \tag{2}$$

- Mann iteration:

$$u_{n+1} = \alpha_n w(u_n) + (1-\alpha_n)u_n, \quad n = 0,1,2,\ldots, \tag{3}$$

  where $\alpha_n \in (0,1]$.

- Ishikawa iteration:

$$\begin{aligned} u_{n+1} &= \alpha_n w(v_n) + (1-\alpha_n)u_n, \\ v_n &= \beta_n w(u_n) + (1-\beta_n)u_n, \quad n = 0,1,2,\ldots, \end{aligned} \tag{4}$$

  where $\alpha_n \in (0,1]$ and $\beta_n \in [0,1]$.

It is easily seen that the Ishikawa iteration with $\beta_n = 0$ for $n = 0,1,2,\ldots$ is Mann iteration, and for $\beta_n = 0$, $\alpha_n = 1$ for $n = 0,1,2,\ldots$ is Picard iteration. The Mann iteration with $\alpha_n = 1$ for $n = 0,1,2,\ldots$ is Picard iteration.

The standard Picard iteration is used in the Banach Fixed Point Theorem [Ber07] to ensure the existence of the fixed point x^* such that $x^* = w(x^*)$ and its approximation under additional assumptions on the space X that should be a Banach one and the mapping w should be contractive. The Mann [Man53] and Ishikawa [Ish74] iterations allow to weak the assumptions on the mapping w.

Our further considerations will be conducted in the space $X = \mathbb{R}^2$ that is obviously a Banach one. We take $u_0 = [x_0, y_0]^T \in \mathbb{R}^2$ and $\alpha_n = \alpha$, $\beta_n = \beta$, such that $\alpha \in (0,1]$ and $\beta \in [0,1]$.

3 NEWTON–RAPHSON METHOD AND ITS GENERALIZATIONS FOR TWO EQUATIONS WITH TWO UNKNOWNS

By square systems we understand systems with as many equations as variables. Take the following system of non-linear equations:

$$\begin{cases} f(x,y) = 0, \\ g(x,y) = 0, \end{cases} \tag{5}$$

where $f, g : \mathbb{R}^2 \to \mathbb{R}$ and x, y are variables.

System (5) can be represented in the form of a single vector equation:

$$\mathbf{F}(x,y) = \begin{bmatrix} f(x,y) \\ g(x,y) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}. \tag{6}$$

We use bold symbols to denote vectors. Assume that $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$ is a continuous function that has continuous first partial derivatives with respect to x and y. To solve equation $\mathbf{F}(x,y) = \mathbf{0}$ one can use the Newton–Raphson method [CK08, SF10] starting from an initial point $\mathbf{z}_0 = [x_0, y_0]^T$:

$$\mathbf{z}_{n+1} = \mathbf{z}_n - \mathbf{J}^{-1}(\mathbf{z}_n)\mathbf{F}(\mathbf{z}_n), \quad n = 0,1,2,\ldots, \tag{7}$$

where

$$\mathbf{J}(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) & \frac{\partial f}{\partial y}(x,y) \\ \frac{\partial g}{\partial x}(x,y) & \frac{\partial g}{\partial y}(x,y) \end{bmatrix} \tag{8}$$

is the 2×2 Jacobian matrix of $\mathbf{F}$ and $\mathbf{J}^{-1}$ is the inverse matrix to $\mathbf{J}$, that in the case of 2×2 matrix is given by the following formula:

$$\begin{aligned} \mathbf{J}^{-1}(x,y) = &\frac{1}{\frac{\partial f}{\partial x}(x,y)\frac{\partial g}{\partial y}(x,y) - \frac{\partial f}{\partial y}(x,y)\frac{\partial g}{\partial x}(x,y)} \cdot \\ &\cdot \begin{bmatrix} \frac{\partial g}{\partial y}(x,y) & -\frac{\partial f}{\partial y}(x,y) \\ -\frac{\partial g}{\partial x}(x,y) & \frac{\partial f}{\partial x}(x,y) \end{bmatrix}. \end{aligned} \tag{9}$$

Introducing the operator $\mathbf{N}(\mathbf{z}) = \mathbf{z} - \mathbf{J}^{-1}(\mathbf{z})\mathbf{F}(\mathbf{z})$ we can represent the Newton–Raphson method in the following short form:

$$\mathbf{z}_{n+1} = \mathbf{N}(\mathbf{z}_n), \quad n = 0,1,2,\ldots. \tag{10}$$

From this form of Newton–Raphson method we clearly see that the method uses Picard iteration.

Applying the Mann iteration (3) in (10) we obtain the following formula:

$$\mathbf{z}_{n+1} = \alpha \mathbf{N}(\mathbf{z}_n) + (1-\alpha)\mathbf{z}_n, \quad n = 0,1,2,\ldots, \quad (11)$$

where $\alpha \in (0,1]$.

Using the Ishikawa iteration (4) in (10) we get:

$$\begin{aligned} \mathbf{z}_{n+1} &= \alpha \mathbf{N}(\mathbf{v}_n) + (1-\alpha)\mathbf{z}_n, \\ \mathbf{v}_n &= \beta \mathbf{N}(\mathbf{z}_n) + (1-\beta)\mathbf{z}_n, \quad n = 0,1,2,\ldots, \end{aligned} \quad (12)$$

where $\alpha \in (0,1]$ and $\beta \in [0,1]$.

Replacement of the Picard iteration by Mann or Ishikawa iterations leads to the new root finding formulas (11) and (12) that are generalizations of the Newton–Raphson method (7). They produce sequences that if convergent, are convergent to any root of $\mathbf{F}$. This follows from the Hahn–Banach Fixed Point Theorem [Ber07]. Formulas (11) and (12) still produce roots finding sequences but with different character of covergence.

The sequence $\{\mathbf{z}_n\}_{n=0}^{\infty}$ (or orbit of the point $\mathbf{z}_0$) converges or not to a root of $\mathbf{F}$. If the sequence converges to a root $\mathbf{z}^*$ then we say that $\mathbf{z}_0$ is attracted to $\mathbf{z}^*$. A set of all starting points $\mathbf{z}_0$ for which $\{\mathbf{z}_n\}_{n=0}^{\infty}$ converges to $\mathbf{z}^*$ is called the basin of attraction of $\mathbf{z}^*$. Boundaries between basins usually have fractal character due to chaotic behaviour of iteration processes. A good example of such situation can be observed while solving the equation $z^3 - 1 = 0$ in complex plane. Investigations of that case directly led to discovery of Julia and Mandelbrot sets [Man83].

To render polynomiograph for system (5) we can use Algorithm 1. As we noticed earlier the Ishikawa iteration is the most general iteration from the considered set of iterations (Picard, Mann and Ishikawa). So, in the algorithm we use the Ishikawa iteration for the Newton–Raphson method (12), and we denote it by $I_{\alpha,\beta}$. In line 8 of the algorithm we see that we need to determine the colour of the starting point. This could be done in very different ways. In the paper we use two methods: basins of attractions and colouring basing on the iteration (iteration colouring). In the first method to each distinct solution of the system we assign a colour. To determine the colour of the starting point we find the closest solution for the last approximation $\mathbf{z}_{n+1}$ and use its colour. In the second method we have a colourmap (table with colours). Now, to determine the colour of the starting point we take the iteration number for which we have left the while loop and map it to index in the colourmap. To map iterations to indices we used linear interpolation.

Algorithm 1: Rendering of polynomiograph for system of equations

Input: $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$ – left side of system (6), $A \subset \mathbb{R}^2$ – area, N – number of iterations, ε – accuracy, α, β – parameters of Ishikawa iteration $I_{\alpha,\beta}$.
Output: Polynomiograph for the area A.

```
1 for z_0 ∈ A do
2     n = 0
3     while n ≤ N do
4         z_{n+1} = I_{α,β}(z_n)
5         if ‖F(z_{n+1})‖ < ε then
6             break
7         n = n + 1
8     Determine colour for z_0 and print it with the colour
```

4 EXAMPLES

In this section we present some polynomiographs for a system of two equations:

$$\begin{cases} x^3 - y = 0, \\ y^3 - x = 0 \end{cases} \quad (13)$$

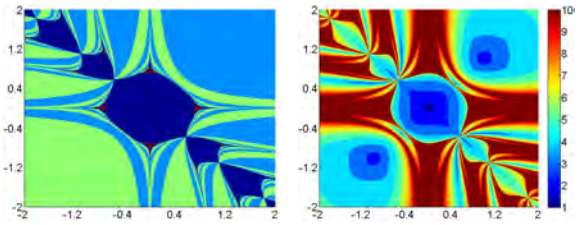
generated using Newton–Raphson method with Picard, Mann and Ishikawa iterations. System (13) has the following four solutions, three of them are real ones and one is complex:

$$\begin{gathered} \{0,0\}, \{1,1\}, \{-1,-1\}, \\ \{0.7071067812 + 0.7071067812i, \\ -0.7071067812 + 0.7071067812i\}. \end{gathered}$$

The common parameters used in the rendering algorithm for all the examples were the following: $A = [-2,2]^2$, $N = 10$, $\varepsilon = 0.001$. The number of points generated in A to obtain the images was set to 600 in each direction.

The examples start with the polynomiographs for the standard Newton–Raphson method, i.e., method with Picard iteration. Fig. 1 presents obtained images. In Fig. 1(a) we see basins of attraction, and in Fig. 1(b) polynomiograph rendered with the iteration colouring (the colourmap is drawn on the right). From the example we see that using the same iteration but different colouring methods we can obtain diverse patterns of polynomiographs.

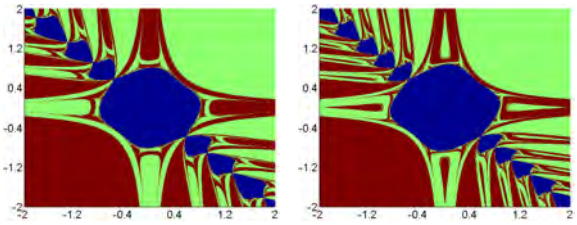
In the second example we use the Mann iteration in the Newton–Raphson method. Fig. 2 presents examples obtained using the basins of attraction colouring method with the following values of α parameter in the Mann iteration: (a) 0.7, (b) 0.5, (c) 0.3, (d) 0.1. Examples showing the use of Mann iteration with iteration colouring are presented in Fig. 3. The values of



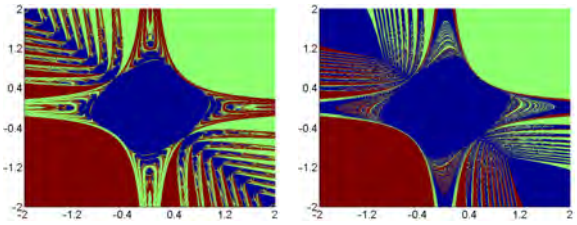
(a) basins of attraction (b) iteration colouring

Figure 1: Polynomiographs for system (13) using Picard iteration.

the α parameter were the following: (a) 0.9, (b) 0.8, (c) 0.7, (d) 0.6. In both cases we see that with the change of α the shape of the polynomiograph changes and their shape is different from the polynomiographs obtained with the Picard iteration (Fig. 1). More interesting changes are noticeable in the case of iteration colouring.



(a) $\alpha = 0.7$ (b) $\alpha = 0.5$

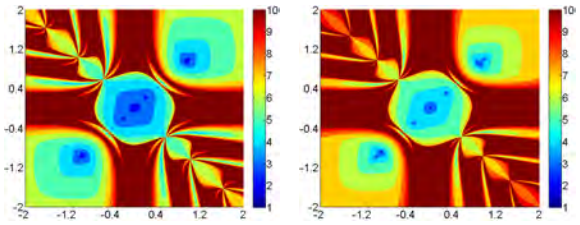


(c) $\alpha = 0.3$ (d) $\alpha = 0.1$

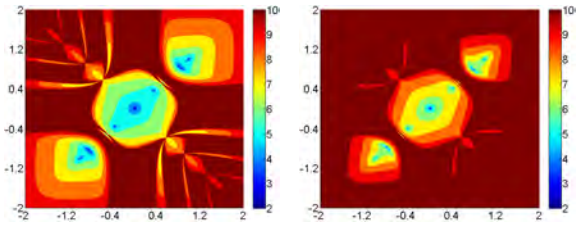
Figure 2: Basins of attraction for system (13) using Mann iteration.

The last example presents the use of Ishikawa iteration in the Newton–Raphson method for system (13). Similar to the case of Mann iteration we generated polynomiographs using two different colouring methods: basins of attraction (Fig. 4) and iteration colouring (Fig. 5). In Fig. 4 we used the following values of the parameters: (a) $\alpha = 0.2$, $\beta = 0.8$, (b) $\alpha = 0.3$, $\beta = 0.7$, (c) $\alpha = 0.7$, $\beta = 0.3$, (d) $\alpha = 0.8$, $\beta = 0.2$, and in Fig. 5 the values were following: (a) $\alpha = 0.6$, $\beta = 0.1$, (b) $\alpha = 0.6$, $\beta = 0.7$, (c) $\alpha = 1.0$, $\beta = 0.7$, (d) $\alpha = 0.7$, $\beta = 0.3$. From the obtained images we clearly see that when we change the parameters values of the Ishikawa iteration we are able to generate a variety of interesting patterns different from those generated with the standard Picard iteration.

Moreover, looking at Fig. 1(a) and Figs. 2, 4 we can observe that the basins of attraction for each of the three

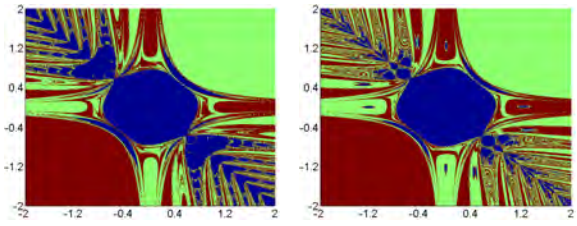


(a) $\alpha = 0.9$ (b) $\alpha = 0.8$

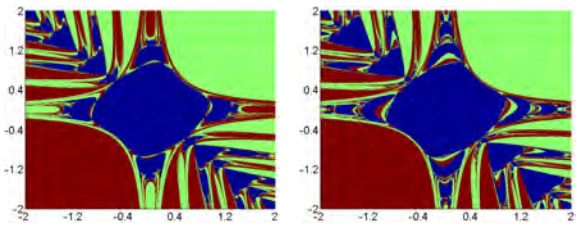


(c) $\alpha = 0.7$ (d) $\alpha = 0.6$

Figure 3: Polynomiographs for system (13) using Mann iteration and iteration colouring.



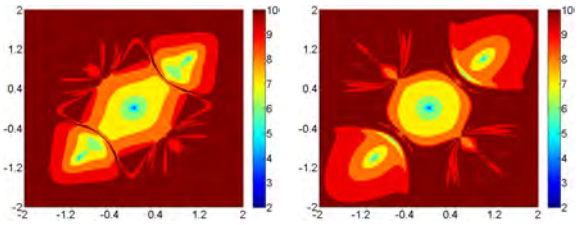
(a) $\alpha = 0.2$, $\beta = 0.8$ (b) $\alpha = 0.3$, $\beta = 0.7$



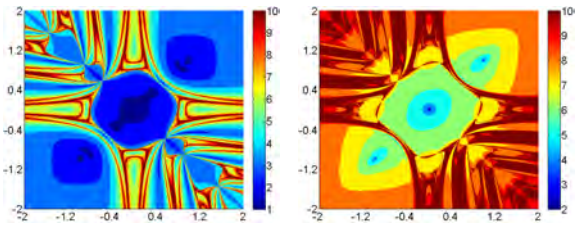
(c) $\alpha = 0.7$, $\beta = 0.3$ (d) $\alpha = 0.8$, $\beta = 0.2$

Figure 4: Basins of attraction for system (13) using Ishikawa iteration.

real roots have significantly changed. Some of them have enlarged and other have been divided into many smaller areas, e.g., Fig. 2(d). Thus, using different values of iterations' parameters for some starting points we are able to converge to different roots. Now, looking at Fig. 1(b) and Figs. 3, 5 we can observe how fast the algorithm has found the roots – speed of convergence. The more red colour in the polynomiograph the slower the algorithm. In most of the cases the convergence of the algorithm with the use of Mann and Ishikawa iteration was slower than with the use of Picard iteration. But there are also cases where we see that for the Mann and Ishikawa iteration the red areas in comparison to the Picard iteration have shrunk and the blue areas have become more darker, e.g., Fig. 5(c), so the speed of convergence is faster. Generally, the change of



(a) $\alpha = 0.6, \beta = 0.1$ (b) $\alpha = 0.6, \beta = 0.7$



(c) $\alpha = 1.0, \beta = 0.7$ (d) $\alpha = 0.7, \beta = 0.3$

Figure 5: Polynomiographs for system (13) using Ishikawa iteration and iteration colouring.

speed depends on the iteration used and the value of its parameters.

5 CONCLUSIONS AND FUTURE WORK

In the paper we presented some generalizations of the classic Newton–Raphson method using Mann and Ishikawa iterations instead of Picard iteration. These generalizations were then applied to a root finding process for square systems of two equations with two unknowns. Obtained different polynomiographs show a great variety of basins of attractions and images presenting speed of convergence for different iterations.

The results of the paper can be further modified in many directions by the usage of multiparameter iterations, different convergence criteria, different colour maps as e.g. in [GKL14, GKL15]. We can also use other colouring methods or other rendering algorithms of polynomiographs, e.g. algorithms presented in [Gda14]. Moreover, we can try to extend the ideas of the paper related to the use of different iterations, visualization methods of the solution finding process to systems with any number of equations and variables.

We believe that results of the paper can be interesting for those whose work or hobbies are related to automatically creating nicely looking graphics. Also we think that they can be applied to increase functionality of existing polynomiography software.

6 REFERENCES

[Ber07] Berinde, V.: Iterative Approximation of Fixed Points, 2nd edn. Springer, Heidelberg (2007)

[CK08] Cheney, W., Kincaid, D.: Numerical Mathematics and Computing, 6th edn. Brooks/Cole, Pacific Groove, CA (2007)

[Gda14] Gdawiec, K.: Mandelbrot- and Julia-like Rendering of Polynomiographs. In: Chmielewski, L.J., et al. (eds.) ICCVG 2014. LNCS, vol. 8671, pp. 25-32. Springer International Publishing (2014)

[GKL14] Gdawiec, K., Kotarski, W., Lisowska, A.: Polynomiography with Non-Standard Iterations. In: WSCG 2014 Poster Proceedings, pp. 21–26 (2014)

[GKL15] Gdawiec, K., Kotarski, W., Lisowska, A.: Polynomiography Based on the Nonstandard Newton-Like Root Finding Methods. Abstract and Applied Analysis, vol. 2015, Article ID 797594, 19 pages (2015)

[Ish74] Ishikawa, S.: Fixed Points by a New Iteration Method. Proceedings of the American Mathematical Society 44(1), 147–150 (1974)

[Kal05a] Kalantari, B.: Polynomiography: From the Fundamental Theorem of Algebra to Art. Leonardo 38(3), 233–238 (2005)

[Kal05b] Kalantari, B.: Two and Three-dimensional Art Inspired by Polynomiography. In: Proceddings of Bridges, Banff, Canada, pp. 321–328 (2005)

[Kal08] Kalantari, B.: Polynomial Root-Finding and Polynomiography. World Scientific, Singapore (2009)

[Man83] Mandelbrot, B.: The Fractal Geometry of Nature. W.H. Freeman and Company, New York (1983)

[Man53] Mann, W.R.: Mean Value Methods in Iteration. Proceedings of the American Mathematical Society 4, 506–510 (1953)

[SF10] Shiskowski, K.M., Frinkle, K.: Principles of Linear Algebra with Maple. John Wiley & Sons, New York, NY (2010)

Visualization and 3D Printing of Multivariate Data of Biomarkers

Michael C. Thrun
DataBionics, University of Marburg, Hans-Meerwein Str., 35032 Marburg, Germany
mthrun@informatik.uni-marburg.de

Florian Lerch
DataBionics University of Marburg, Hans-Meerwein Str., 35032 Marburg, Germany
lerchf@students.uni-marburg.de

Jörn Lötsch[1]
Institute of Clinical Pharmacology, Goethe - University, Theodor Stern Kai 7, 60590 Frankfurt am Main, Germany
j.loetsch@em.uni-marburg.de

Alfred Ultsch
DataBionics University of Marburg, Hans-Meerwein Str., 35032 Marburg, Germany
ultsch@mathematik.uni-marburg.

(1) Fraunhofer Institute for Molecular Biology and Applied Ecology IME, Project Group Translational Medicine and Pharmacology TMP, Theodor-Stern-Kai 7, 60590 Frankfurt am Main, Germany

ABSTRACT

Dimensionality reduction by feature extraction is commonly used to project high-dimensional data into a low-dimensional space. With the aim to create a visualization of data, only projections onto two dimensions are considered here. Self-organizing maps were chosen as the projection method, which enabled the use of the U*-Matrix as an established method to visualize data as landscapes. Owing to the availability of the 3D printing technique, this allows presenting the structure of data in an intuitive way. For this purpose, information about the height of the landscapes is used to produce a three dimensional landscape with a 3D color printer. Similarities between high-dimensional data are observed as valleys and dissimilarities as mountains or ridges. These 3D prints provide topical experts a haptic grasp of high-dimensional structures. The method will be exemplarily demonstrated on multivariate data comprising pain-related bio responses. In addition, a new R package "Umatrix" is introduced that allows the user to generate landscapes with hypsometric tints.

Keywords

Self-Organizing Map (SOM), Multivariate Data Visualization, Dimensionality Reduction, High Dimensional Data, 3D Printing, U-Matrix.

1. Introduction

Some large data sets possess a high number of variables with a low number of observations. Projection methods reduce the dimension of the data and try to represent structures present in the high dimensional space. If the projected data is two dimensional, the positions of projected points do not represent high-dimensional distances. Therefore, low dimensional similarities could lead to incorrect interpretations of the underlying structures.

A certain solution for this problem is the self-organizing map (SOM) [Kohonen, 1982] with high number of neurons used as a projection method [Ultsch, 1999]. SOM is an unsupervised neural learning algorithm. If used as a projection method, the picture of high-dimensional data is uniformly distributed on the neural grid. This distribution makes a direct interpretation demanding. The standard approach for this problem lies in generating a 2D visualization for SOM, because, for high-dimensional data, the SOM remains a reference tool for 2D visualizations [Lee/Verleysen, 2007, p. 227]. In literature, there are many approaches, which require experienced interpretations (e.g.[Kadim Tasdemir/Merényi, 2012; Vesanto/Alhoniemi, 2000]). Here, we focus on the method of U*matrix, which is able to visualize distance and density based structures. The U*matrix leads to a topographic map with hypsometric tints (for details see section 5), which seems like a 3D landscape for the human eye. But every 3D visualization still has to be viewed from multiple viewpoints and is often subject to serious occlusion, distortion and navigation issues [Jansen et al., 2013] cites [Shneiderman, 2003]. But [Jansen et al., 2013] showed that physical

visualizations can improve the user's efficiency at information retrieval tasks, because physical touch seems to be an essential cognitive aid. Three-dimensional printing addresses this important point through generating a haptic form. To facilitate this visualization of high-dimensional data for experts in the data's field, we propose the usage of colored three-dimensional printing.

3D printing is currently a quickly evolving technique. It represents a technical change from spraying toner on paper to adding up layers of materials to a 3D object [Sachs et al., 1993]. By enabling a machine to produce objects of any shape it has the potential to impact many production areas [D'aveni, 2013]. Main biomedical applications were so far 3D printing vascular implants, aerosol delivery technologies, cellular transplantation, endo-prosthetics, tissue engineering, biomedical device development and pharmacology including techniques such as individualized drug delivery formulations [Pillay/Choonara, 2015]. 3D printing is also employed for the visualization of biomedical data, for example to produce graspable three-dimensional objects for surgical planning [Rengier et al., 2010].

This work proposes the application of 3D printing to the enhancement of knowledge discovery in high-dimensional data transferring them into 3D haptic physical models with the goal of physical grasping a visualization of projections.

The results are shown using the example of pain data. Blue and green valleys indicate clusters of pain types and the brown or white watersheds of the U*matrix point to borderlines of clusters (Fig. 4). Other SOM visualizations fail to display the information in an easily understandable form and do not allow the usage of 3D printing (see section 3).

We enable the user to achieve every step until the 3D printing using software: The tasks of SOM generation, visualization and supervised clustering can be performed interactively by the R package Umatrix [Version 2.0.0; Thrun et al., 2016]. The package also enables the usage of other SOM algorithms or comparing classifications with the U*matrix visualization.

2. Emergent SOM

The first step for structure visualization is to project high-dimensional data in a two dimensional space. One approach is using self-organizing maps (SOM), which project to a fixed grid of neurons. Originally, the SOM algorithm was introduced by [Kohonen, 1982]. However, to exploit emergent phenomena in SOMs [Ultsch, 1999] argued to use a large number of neurons (at least $n = 4000$). The self-organization of many neurons allows emergent structures to occur in data. By gaining the property of emergence through self-organization this enhancement of SOM is called Emergent SOM (ESOM).

Let $M = \{m_1, \dots, m_n\}$ be the positions of neurons on a two dimensional grid (map) and $W = \{w(m_i) = w_i | i = 1, \dots n\}$ the corresponding set of weights or prototypes of neurons, then the SOM learning algorithm constructs a nonlinear and topology preserving projection of the input space I by finding the bestmatching unit (BMU):

$$BMU(l) = \underset{m_i \in M}{\operatorname{argmin}}\{D(l, w_i)\}, \quad i \in \{1, \dots, n\} \quad (1)$$

$\forall l \in I$, if D denotes a distance between input space I. Hence, the location of a given data point on the resulting map is depicted by the corresponding BMU. The topology of the map is toroid if the borders are cyclically connected [Ultsch, 1999]. If the map was planar, the neighborhood of neurons at the edges would contain much less neurons compared to the middle of the map space. This would lead to undesired seam effects in the SOM algorithm [Ultsch, 2003a].

In each step the SOM learning is achieved by modifying the weights in a neighborhood with

$$\Delta w(R) = \eta(R) * h(BMU(l), m_i, R) * (l - w(m_i)) \quad (2).$$

The cooling scheme is defined by the neighborhood function $h: M \times M \times \mathbb{R}^+ \to [-1,1]$ and the learning rate η: $\mathbb{R}^+ \to [0,1]$, where the radius R declines until $R = 1$ through the definition of the maximum number of epochs.

3. Other visualizations of SOMs

The result of Kohonen SOM algorithm are neurons, which are located on a map with a set W of prototypes corresponding to a set M of positions. In general, the positions on M are restricted to a grid, but a few approaches exist which change the positions in M, like Adaptive Coordinates [Merkl/Rauber, 1997]. Because these approaches are not based on a grid, they are not considered further.

BMUs define locations of input points on the map. However, they exhibit no structure of the input space for a SOM [Ultsch, 1999]. But the goal is to grasp the structure of the high dimensional data and maybe even visualize cluster boundaries. Therefore, post-processing of the neurons is required for an informative representation of high dimensional data. Three standard approaches are found in literature:

The first approach projects the prototypes of the set W with Multidimensional Scaling (MDS) [Torgerson, 1952] or some of its variants to a two dimensional space [Kaski et al., 2000; Sarlin/Rönnqvist, 2013]. The result is mapped into the CIELab color space [Colorimetry, 2004]. This uniform color space is defined so that perceptual differences in colors

correspond to Euclidean distances in the map space as well as possible [Kaski et al., 2000]. The next two approaches visualize either distances or density of the prototypes.

The second approach defines receptive fields around each position in *M*. The unified distance matrix (Umatrix) [Ultsch/Siemon, 1990] or variants [Kraaijveld et al., 1995] [Häkkinen/Koikkalainen, 1997] [Hamel/Brown, 2011] represent distances of prototypes (see section 4 for details) by using proportional intensities of gray shades, color hues, shape or size. In [Kraaijveld et al., 1995] every neuron corresponds to a pixel. The gray value of each pixel is determined by the maximum unit distance from the neuron to its four neighbors (up, down, left, right). The larger the distance, the lighter the gray value. In [Häkkinen/Koikkalainen, 1997] additional visualization approaches for unit distances are explained. The shape and size of the receptive fields describe the dissimilarity of the corresponding neurons. Apart from the U-matrix, visualizations of receptive fields in three dimensions or specific components of prototypes with receptive fields in two dimensions were tried [Vesanto, 1999]. Also, SOM quality measures can be added to the receptive fields in a third dimension, e.g. [Vesanto et al., 1998].

The third approach connects the positions *M* by way of a specific scheme. In [Hamel/Brown, 2011] additional to a U-matrix neurons are connected with lines along the maximum gradient. The authors claim that clusters are the always connected components of the graph defined by the Umatrix .

[Merkl/Rauber, 1997] omitted the receptive fields approach by only connecting map positions with lines, where the intensity of the connections reflects the similarity of the underlying prototypes. [K. Tasdemir/Merenyi, 2009] proposed the CONNvis technique, which visualizes the grid by connecting the neurons, whose corresponding prototypes are adjacent in the space of input dimensionality, which is equal to the high dimensional data. The width of the connection line is proportional to the strength of the connection [K. Tasdemir/Merenyi, 2009].

In sum, all visualizations of large SOMs described above require an expert in the field for interpretation. In addition, a 3D print may not give a desirable result: in most cases the 2D visualization would have to be enhanced to 3D. But research indicates that 3D does not improve 2D visualizations [Cockburn, 2004; Cockburn/McKenzie, 2002; Sebrechts et al., 1999], and, to our knowledge, there are no 3D visualizations of ESOMs based on a 2D grid currently in use, besides the approach proposed in section 5.

4. U*matrix based on data distances and density

The Umatrix displays a folding of high-dimensional space, where each receptive field is called a U-height. Let *N(j)* be the eight immediate neighbors of $m_j \in M$, let $w_j \in W$ be the corresponding prototype to m_j, then the average of all distances between prototypes w_i is called U-height regarding the position m_j:

$$u(j) = \frac{1}{n}\sum_{i \in N(j)} D(w_i, w_j)\,,\ n = |N(j)| \quad (3).$$

The Umatrix is a display of proportional intensities of grey shades of all receptive fields [Ultsch, 2003a]. By formalizing the displayed structures [Lötsch/Ultsch, 2014] showed that the Umatrix is an approximation of Voronoi borders of the high-dimensional points in the output space:

Let *bmu(l)* and *bmu(j)* be BMUs of data points *l* and *j*, where *bmu(j)* and *bmu(l)* have bordering Voronoi cells. On the borderline there is a vertical plane (AU-height), which is the distance *D(l,j)* > 0 between the data points in the input space. In sum, the abstract Umatrix, (AU-matrix) is the Delaunay graph of the BMU's weighted by corresponding Euclidean distances in the input space.

In addition to the Umatrix, [Ultsch, 2003a] introduced the high-dimensional density visualization technique called P-Matrix, where P-heights on top of the receptive fields are displayed. The P-height $p(m_i)$ for a position m_i is a measure of the density of data points in the vicinity of $w(m_j)$:

$$p(m_j) = |\{i \in I | D(i, w(m_j)) < r > 0, r \in \mathbb{R}\}| \quad (4).$$

The P-height is the number of data points within a hypersphere of radius *r*. Here, we choose the interval ϱ of the radius with

$$\varrho \in [median(C(D)), median(A(D))], \quad (5)$$

where *D* are all input space distances and *A(D)* is the group *A* of distances calculated by the ABCanalysis [Ultsch/Lötsch, 2015]. ABCanalysis tries to identify the optimum information that can be validly retrieved by using concepts developed in economical sciences. In particular, concepts are used in the search for a minimum possible effort that gives the maximum yield [Ultsch/Lötsch, 2015]. The distances are divided into three disjoint subsets *A*, *B* and *C*, with subset *A* comprising largest values ("outer cluster distances"), subset *B* comprising values where the yield equals the effort required to obtain it, and the subset *C* comprising of the smallest values ("inner cluster distances"). We suggest the choice for the specific radius *r* through the proportion *v* of inter- versus intra-cluster distances estimated by

$$v = \frac{max(C(D))}{min(A(D))} \quad (6).$$

The radius r is estimated by $r = v * p20(D)$, where $p20(D)$ is 20-th percentile of input distances [Ultsch, 2003b]. From this starting point the user may search interactively for the empirical Pareto percentile, which defines the radius *r* (see R package Umatrix).

The combination of a Umatrix and a Pmatrix is called U*matrix [Ultsch et al., 2016]: It can be formalized as pointwise matrix multiplication: $U^* = U * F(P)$, where *F(P)* is a matrix of factors *f(p)* that are determined through a linear function *f* on the *P* heights *p* of the Pmatrix. The function *f* is calculated so that *f(p) = 1* if *p* is equal to the median and *f(p) = 0* if *p* is equal to the 95-percentile ($p95$) of the heights in the Pmatrix. For *p(j) > p95: f(p) = 0*, which indicates that *j* is well within a cluster and results in zero heights in the U*matrix.

5. Visualization as a 3D landscape

We concur with [Koikkalainen, 1997] that the content of information should be displayed in an understandable way. Hence, in the following section we formalize the idea of [Ultsch, 2003a] to visualize the U*matrix as a landscape. We define a topographic map with hypsometric tints [Patterson/Kelso, 2004]. Hypsometric tints are surface colors which depict ranges of elevation. Here, a specific color scale is combined with contour lines.

The color scale is chosen to display various valleys, ridges and basins: blue colors indicate small distances (sea level), green and brown colors indicate middle distances (small hilly country) and white colors indicate high distances (snow and ice of high mountains). The valleys and basins indicate clusters and the watersheds of hills and mountains indicate borderlines of clusters (Fig. 1 and Fig. 4).

The landscape consists of receptive fields, which correspond to intervals of U*heights edged by contours. This paper proposes the following approach: First, the range of U*heights is assigned uniformly and continuously to the specific color scale above by robust normalization [Milligan/Cooper, 1988] and by splitting it up into intervals. In the next step, the color scale is interpolated by the corresponding CIELab colors space [Colorimetry, 2004]. The largest possible contiguous areas of receptive fields, which are in the same U*height interval, are summarized and outlined in black as a contour. In sum, a receptive field is the display of one color in one particular place of the U*matrix visualization within a height dependent contour. Let *u(j)* be the U*height, *q01* the one-percentile and *q99* the 99-percentile of U*heights, then the robust normalization of U*heights *u(j)* is defined by

$$u(j) = \frac{u(j) - q01}{q99 - q01} \quad (7).$$

The number of intervals *in* is defined by

$$\frac{1}{in} = \frac{q01}{q99}. \quad (8).$$

The resulting visualization consists of a hierarchy of areas of different height levels with corresponding colors (see Fig. 4). The visualization of SOMs using the tool Umatrix is consistent with a 3D landscape for the human eye, therefore one sees data structures intuitively. Contrary to other SOM visualizations, e.g. [K. Tasdemir/Merenyi, 2009], the 3D landscape enables layman to interpret the results of a SOM.

Using a toroid map for the ESOM computation requires a tiled display of the landscape in the interactive tool Umatrix [Version 2.0.0; Thrun et al., 2016] which means that every receptive field is shown four times. So in the first step the visualization consists of four adjoining pictures of the same Umatrix [Ultsch, 2003a] (the same for the U*matrix after loading of a SOM or computing one). To get the 3D landscape this paper proposes to cut the tiled U*matrix visualization rectangular:

Let v_{Lines} be the vector of row sums, $v_{Columns}$ be the vector of column sums of the U*heights and let b_{Lines} be the number of BMU's of the corresponding row line of v_{Lines} (for $b_{Columns}$, $v_{Columns}$), then we define the upper border $\text{up} = \max(v_{Lines}/f(b_{Lines}))$, the left border by $\text{lb} = \max(b_{Columns}/f(v_{Columns}))$ and the other two borders by the length and width of the U*matrix, if the vector *f(b)* is the addition $f(b) = \hat{b} + b + \check{b}$ with $\hat{b} = (b_n, b_1, \dots, b_{n-1})$ and $\check{b} = (b_2, \dots, b_{n+1})$, where the grid is toroid. For better comprehensibility see the axes in Fig 1, which are defined from one to $max(Lines)$ and from one to $max(Columns)$.

6. 3D Printing of pain phenotypes

3D landscapes can be better grasped when viewed from multiple perspectives. This can be easily achieved with a haptic form. As an example of a haptic 3D presentation of biomedical data, complex pain phenotypes composed of responses to four different types of nociceptive stimuli are used. Nociceptive stimuli activate nociceptors, which are sensory nerve cells responding to pressure (mechanic), electric, cold or heat. Data was acquired with the help of 206 healthy volunteers as described previously in detail [Flühr et al., 2009; Lötsch/Ultsch, 2013; Neddermeyer et al., 2008]. Data was projected using the ESOM algorithm and clusters were identified by interpreting its U*matrix visualization (Fig 1). In a last step, pain sub-phenotypes were identified by interpreting the clusters using classification and regression tree classifiers (Cart) [Lötsch/Ultsch, 2013]. By way of extracting decision rules through the conditional information of the GINI impurity [Hill et al., 2006], the interpretation based on measured stimulus intensities evoking pain at threshold level. Eight

different pain phenotypes were observed, involving individuals who shared complex pain threshold patterns across five variables. Subsequently, the specific properties of each phenotype could be interpreted clinically. Three main pain sensitivity groups were identified: high-pain sensitivity (HPS), average pain sensitivity (APS) and low-pain sensitivity (LPS) [Lötsch/Ultsch, 2013]. HPS was divided into two clusters (1,2), APS into four (3-6) and LPS into two (7,8). All clusters were interpretable (further details see [Lötsch/Ultsch, 2013]). From this data set, a 3D Landscape could be generated (Fig. 1 top view and Fig. 4) and printed by means of a 3D color printer (Fig. 2). Due to technical limitations, printing is restricted to three colors blue, green and white, while the digital 3D landscape consists of many more different height dependent colors (Fig. 2. and Fig. 4). On the other hand, contrary to Fig 1, Fig. 4 had to be reworked manually by using a graphics editor program. Otherwise the structures on the borders of the island would be difficult to interpret. Note, that the 3D print of Fig. 2 was generated using Fig. 1 and not Fig 4.

Data processing was done using the interactive tool Umatrix [Version 2.0.0; Thrun et al., 2016] with the freely available R software [Version 3.2.5; R Development Core Team, 2008] for Windows 7 64bit, and the graphical interface by the open source web application framework shiny [Version 0.13.2; RStudio, 2014]. To our knowledge, the 3D print of an U*matrix is the first application of 3D printing techniques used directly for data mining and knowledge discovery in high-dimensional data in a haptic form. In addition, the political map of the eight clusters is shown in Fig 3. The political map of an ESOM is the coloring of the Voronoi cells of the BMUs with different colors for each cluster [Lötsch/Ultsch, 2014].

7. Summary

Projection methods visualize the structures of high-dimensional data in a low-dimensional space. The unsupervised neural learning algorithm, which is called self-organizing map (SOM), may be used as a non-linear projection method. In that case SOM projects high-dimensional data onto a two dimensional grid, where the positions of projected points do not represent high-dimensional distances. The standard approach to this problem is the generation of a visualization for SOM. Because common SOM visualizations fail to display the information in an easily understandable form and do not allow the usage of 3D printing, we combined a large SOM with the U*matrix visualization technique. The U*matrix is able to visualize distance and density based structures. This 3D visualization is a topographic map with hypsometric tints and representable as a 3D landscape. The details of creating the 3D landscape were introduced in the paper in section 5. The tasks of SOM generation, visualization and supervised clustering can be performed interactively by the published R package Umatrix [Version 2.0.0; Thrun et al., 2016]. We allow the user to choose a different SOM based projection method, on which our visualization techniques still can be used. The package also enables comparing of classifications to the U*matrix visualization.

The main step forward presented in this paper is the color 3D printing of landscapes based on the visualization originating from the U*matrix. Through its haptic form, the 3D print makes high-dimensional structures more understandable for experts in the data's field. Structural features of high-dimensional data were depicted with the use of 3D printing (Fig 2) and pain data. Blue and green valleys indicate clusters of pain types and the brown or white watersheds of the U*matrix visualization point to borderlines of clusters. In our opinion, the task of height depending 3D color printing is still very trying. Automatically cutting a non-rectangular island defined by curved borders remains also an unsolved problem.

To our knowledge, this 3D print is the first application of 3D printing techniques used directly for data mining and knowledge discovery in high-dimensional data in a haptic form.

Future work will include the abstract U*matrix [Ultsch et al., 2016] into the current visualization techniques and allow the height dependent 3D print of an U*matrix in more than three colors.

8. Acknowledgments

This work has been funded by the Landesoffensive zur Entwicklung wissenschaftlich-ökonomischer Exzellenz (LOEWE), LOEWE-Zentrum für Translationale Medizin und Pharmakologie (JL), by the German Research Foundation (DFG) under grant agreement (BE4234/3-1, UL159/10-1), and by the Else Kröner-Fresenius Foundation (EKFS), Research Training Group Translational Research Innovation – Pharma (TRIP, JL). Special acknowledgment goes to the 3D printing by Michael Weingart, Weingart Ingenieur-Büro + CNC Fräsen, Kirchheim-Teck, Germany for the practical 3D print and the consistent coloring of the print.

9. References

Cockburn, A.: Revisiting 2D vs 3D implications on spatial memory, Proc. Proceedings of the fifth conference on Australasian user interface-Volume 28, pp. 25-31, Australian Computer Society, Inc., 2004.

Cockburn, A., & McKenzie, B.: Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments, Proc.

Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 203-210, ACM, 2002.

Colorimetry.C.I.E., Vol. CIE Publication,Central Bureau of the CIE, Vienna, 2004.

D'aveni, R. A.: 3-D printing will change the world, Harvard business review, Vol. 91(3), pp. 34-35. 2013.

Flühr, K., Neddermeyer, T. J., & Lötsch, J.: Capsaicin or menthol sensitization induces quantitative but no qualitative changes to thermal and mechanical pain thresholds, The Clinical journal of pain, Vol. 25(2), pp. 128-131. 2009.

Häkkinen, E., & Koikkalainen, P.: SOM based visualization in data analysis, Artificial Neural Networks—ICANN'97, (pp. 601-606), Springer, 1997.

Hamel, L., & Brown, C. W.: Improved interpretability of the unified distance matrix with connected components, Proc. 7th International Conference on Data Mining (DMIN'11), pp. 338-343, 2011.

Hill, T., Lewicki, P., & Lewicki, P.: Statistics: methods and applications: a comprehensive reference for science, industry, and data mining, StatSoft, Inc., 2006.

Jansen, Y., Dragicevic, P., & Fekete, J.-D.: Evaluating the efficiency of physical visualizations, Proc. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2593-2602, ACM, 2013.

Kaski, S., Venna, J., & Kohonen, T.: Coloring that reveals cluster structures in multivariate data, Australian Journal of Intelligent Information Processing Systems, Vol. 6(2), pp. 82-88. 2000.

Kohonen, T.: Self-organized formation of topologically correct feature maps, Biological cybernetics, Vol. 43(1), pp. 59-69. 1982.

Koikkalainen, E. H. P.: The neural data analysis environment, Proceedings of the Workshop on Self-Organizing Maps Map, Vol., pp. 69-74. 1997.

Kraaijveld, M., Mao, J., & Jain, A. K.: A nonlinear projection method based on Kohonen's topology preserving maps, Neural Networks, IEEE Transactions on, Vol. 6(3), pp. 548-559. 1995.

Lee, J. A., & Verleysen, M.: Nonlinear dimensionality reduction, New York, USA, Springer, 2007.

Lötsch, J., & Ultsch, A.: A machine-learned knowledge discovery method for associating complex phenotypes with complex genotypes. Application to pain, Journal of biomedical informatics, Vol. 46(5), pp. 921-928. 2013.

Lötsch, J., & Ultsch, A.: Exploiting the Structures of the U-Matrix, in Villmann, T., Schleif, F.-M., Kaden, M. & Lange, M. (eds.), Proc. Advances in Self-Organizing Maps and Learning Vector Quantization, pp. 249-257, Springer International Publishing, Mittweida, Germany, 2014.

Merkl, D., & Rauber, A.: Alternative ways for cluster visualization in self-organizing maps, Proc. Proc. of the Workshop on Self-Organizing Maps (WSOM97), pp. 4-6, Citeseer, 1997.

Milligan, G. W., & Cooper, M. C.: A study of standardization of variables in cluster analysis, Journal of Classification, Vol. 5(2), pp. 181-204. 1988.

Neddermeyer, T. J., Flühr, K., & Lötsch, J.: Principle components analysis of pain thresholds to thermal, electrical, and mechanical stimuli suggests a predominant common source of variance, Pain, Vol. 138(2), pp. 286-291. 2008.

Patterson, T., & Kelso, N. V.: Hal Shelton revisited: Designing and producing natural-color maps with satellite land cover data, Cartographic Perspectives, Vol. (47), pp. 28-55. 2004.

Pillay, V., & Choonara, Y.: 3D Printing in Drug Delivery Formulation: You Can Dream it, Design it and Print it. How About Patent it?, Recent patents on drug delivery & formulation, Vol., pp., 2015.

R Development Core Team. (2008). R: A Language and Environment for Statistical Computing (Version 3.2.5). Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

Rengier, F., Mehndiratta, A., von Tengg-Kobligk, H., Zechmann, C. M., Unterhinninghofen, R., Kauczor, H.-U., & Giesel, F. L.: 3D printing based on imaging data: review of medical applications, International journal of computer assisted radiology and surgery, Vol. 5(4), pp. 335-341. 2010.

RStudio, I. (2014). shiny: Easy web applications in R (Version 0.13.2). Retrieved from http://shiny.rstudio.com/

Sachs, E., Cima, M., Cornie, J., Brancazio, D., Bredt, J., Curodeau, A., . . . Lee, J.: Three-dimensional printing: the physics and implications of additive manufacturing, CIRP Annals-Manufacturing Technology, Vol. 42(1), pp. 257-260. 1993.

Sarlin, P., & Rönnqvist, S.: Cluster coloring of the Self-Organizing Map: An information visualization perspective, arXiv preprint arXiv:1306.3860, Vol., pp., 2013.

Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., & Miller, M. S.: Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces, Proc. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 3-10, ACM, 1999.

Shneiderman, B.: Why not make interfaces better than 3D reality?, Computer Graphics and Applications, IEEE, Vol. 23(6), pp. 12-15. 2003.

Tasdemir, K., & Merenyi, E.: Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps, IEEE Transactions on Neural Networks, Vol. 20(4), pp. 549-562. doi 10.1109/tnn.2008.2005409, 2009.

Tasdemir, K., & Merényi, E.: SOM-based topology visualisation for interactive analysis of high-dimensional large datasets, Machine Learning Reports, Vol. 1, pp. 13-15. 2012.

Thrun, M. C., Lerch, F., & Ultsch, A. (2016). Umatrix (Version 2.0.0). Marburg. R package, requires CRAN packages: Rcpp, ggplot2, shiny, ABCanalysis, shinyjs, reshape2, fields, plyr, abind, tcltk, png, tools, grid, rgl. Retrieved from

www.uni-marburg.de/fb12/datenbionik/software-en

Torgerson, W. S.: Multidimensional scaling: I. Theory and method, Psychometrika, Vol. 17(4), pp. 401-419. 1952.

Ultsch, A.: Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, In Oja, E. & Kaski, S. (Eds.), Kohonen maps, (1 ed., pp. 33-46), Elsevier, 1999.

Ultsch, A.: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, 2003a.

Ultsch, A.Optimal density estimation in data containing clusters of unknown structure, technical report, Vol. 34,University of Marburg, Department of Mathematics and Computer Science, 2003b.

Ultsch, A., Behnisch, M., & Lötsch, J.: ESOM Visualizations for Quality Assessment in Clustering, In Merényi, E., Mendenhall, J. M. & O'Driscoll, P. (Eds.), Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 11th International Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016, (10.1007/978-3-319-28518-4_3pp. 39-48), Cham, Springer International Publishing, 2016.

Ultsch, A., & Lötsch, J.: Computed ABC Analysis for Rational Selection of Most Informative Variables in Multivariate Data, PloS one, Vol. 10(6), pp. e0129767. doi 10.1371/journal.pone.0129767, 2015.

Ultsch, A., & Siemon, H. P.: Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, International Neural Network Conference, pp. 305-308, Kluwer Academic Press, Paris, France, 1990.

Vesanto, J.: SOM-based data visualization methods, Intelligent data analysis, Vol. 3(2), pp. 111-126. 1999.

Vesanto, J., & Alhoniemi, E.: Clustering of the self-organizing map, Neural Networks, IEEE Transactions on, Vol. 11(3), pp. 586-600. 2000.

Vesanto, J., Himberg, J., Siponen, M., & Simula, O.: Enhancing SOM based data visualization, Proc. Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems. Methodologies for the Conception, Design and Application of Soft Computing, Vol. 1, pp. 64-67, 1998.

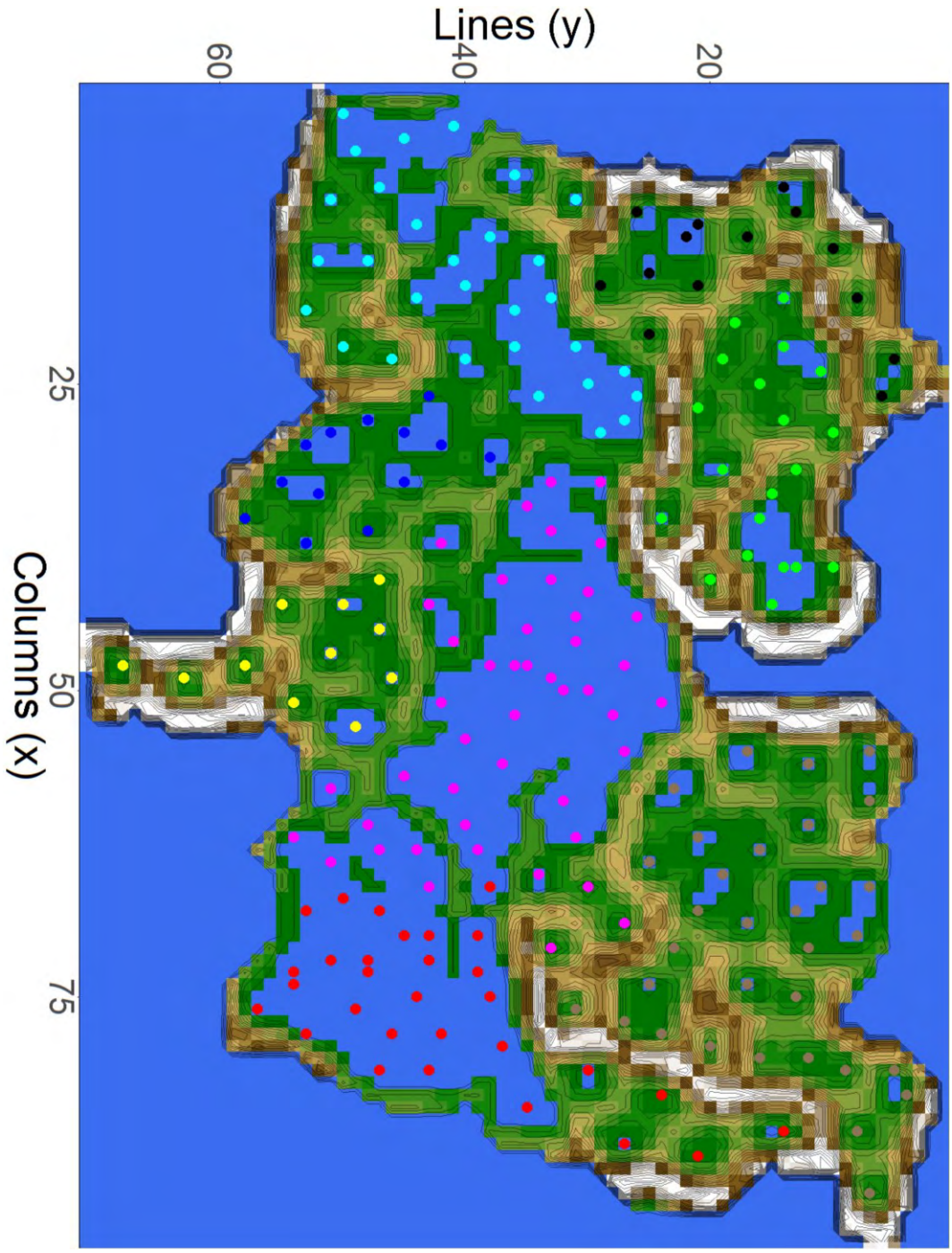


Figure 1: Top view of the 3D landscape of the pain data generated with the Umatrix tool: After the rectangular cut (section 5), the cutting lines of visualization of the U*matrix were improved interactively. The points are the BMU's with different colors as cluster labels. The top view was used for 3D printing.

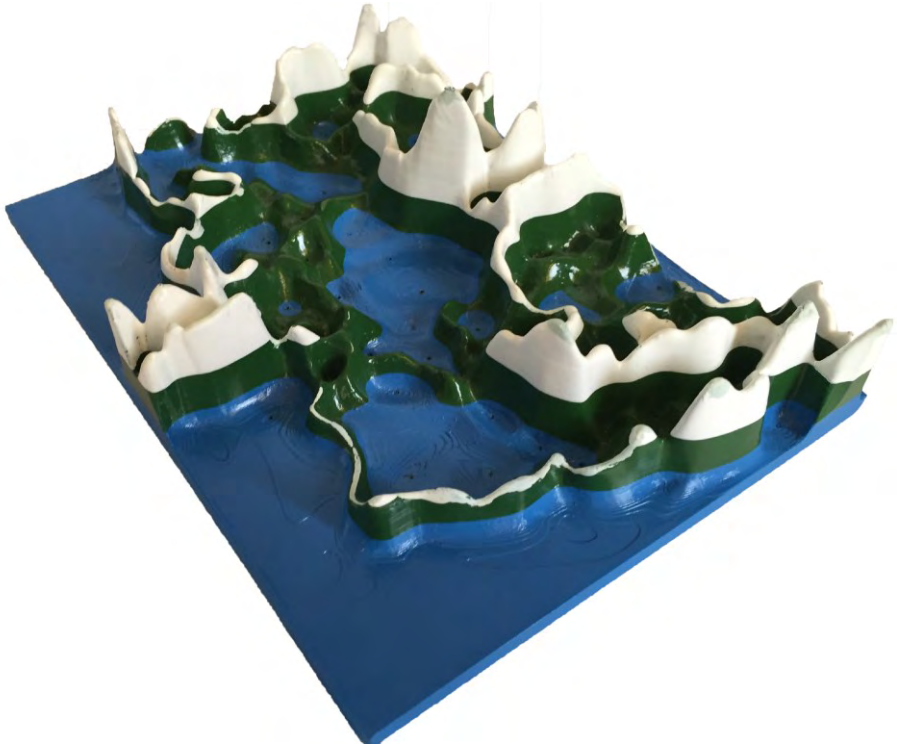


Figure 2: The 3D print of Fig 1 in three height dependent colors white, green and blue. The valleys indicate clusters of pain types and the watersheds of the U*matrix borderlines of clusters.

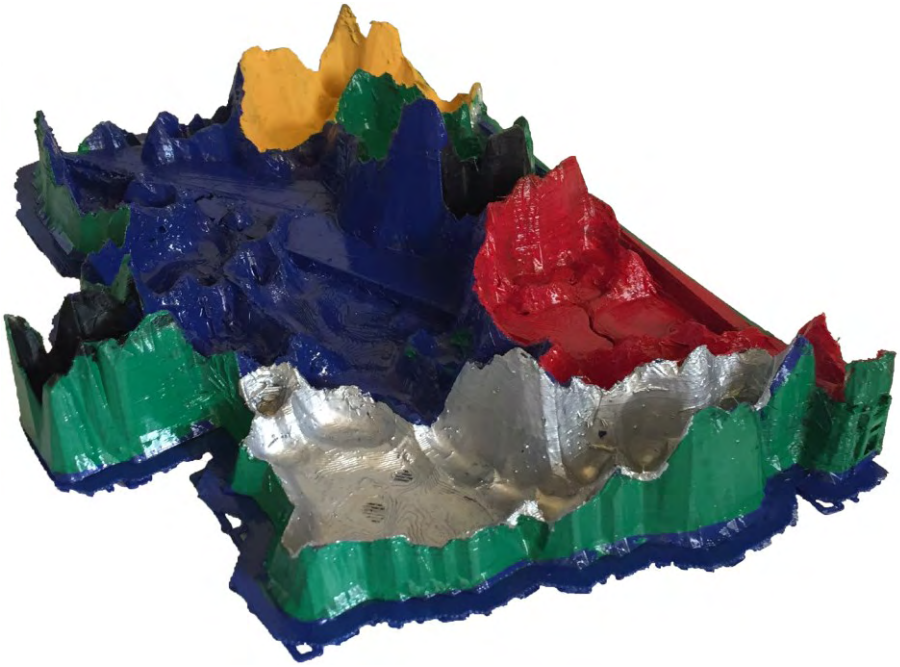


Figure 3: AU*-clustering based on the Voronoi cells formalizes the distance and density based structures and leads from Fig 1 to a political map (further details in [Ultsch et al., 2016]). Above the 3D print of this political map is shown. Every color indicates one cluster as described in section 6.

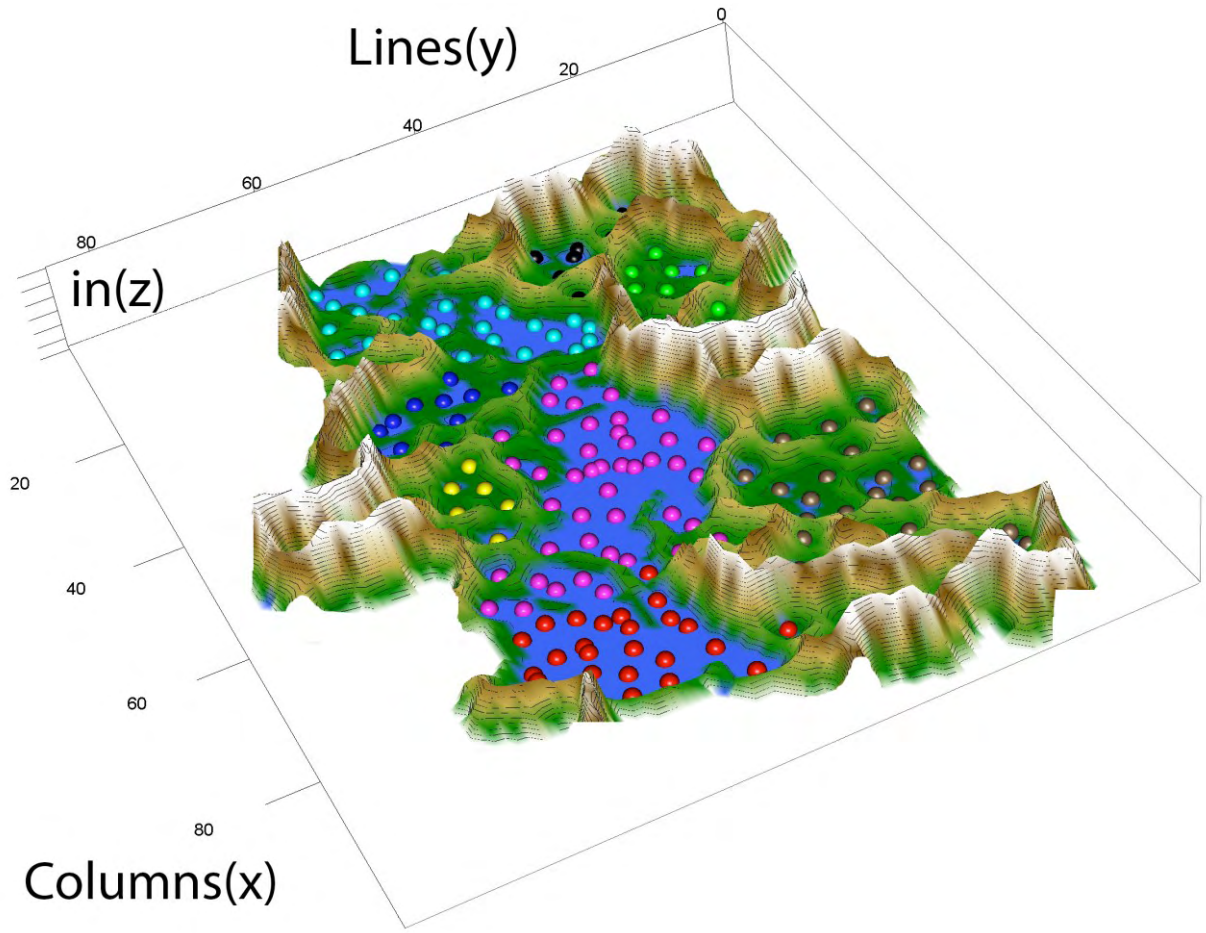


Figure 4: 3D landscape of the pain data generated with the Umatrix tool: After the rectangular cut (section 5), the cutting lines of visualization of the U*matrix were improved interactively with shiny in R. The points are the BMU's with different colors as cluster labels. Contrary to Figure 1, the borders around the island had to be reworked manually using graphics editor program afterwards. Otherwise the borders of the island would be difficult to interpret.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Faking It: Simulating Background Blur in Portrait Photography using a Coarse Depth Map Estimation from a Single Image

Nadine Friedrich Oleg Lobachev Michael Guthe
University Bayreuth, AI5: Visual Computing, Universitätsstraße 30, D-95447 Bayreuth, Germany

Figure 1: Our approach vs. a real image with bokeh. Left: input image, middle: result of our simulation, right: gold standard image, captured with the same lens as the input image, but with a large aperture, yielding natural background blur.

ABSTRACT

In this work we simulate background blur in photographs through a coarse estimation of a depth map. As our input is a single portrait picture, we constraint our objects to humans first and utilise skin detection. A further extension alleviates this. With auxiliary user input we further refine our depth map estimate to a full-fledged foreground–background segmentation. This enables the computation of the actual blurred image at the very end of our pipeline.

Keywords

bokeh, background blur, depth map, foreground–background segmentation

1 INTRODUCTION

High-quality portrait photography often features a special kind of background blur, called bokeh. Its nature originates from the shape of camera lenses, aperture, distance to background objects, and their distinctive light and shadow patterns. This effect is thus used for artistic purposes, it separates the object the lens is focused on from the background and helps the viewer to concentrate on the foreground object—the actual subject of the photograph.

We do not render a depth-of-field blur in a 3D scene, but pursue a different approach. Our input is a single 2D image without additional data—no depth field, no IR channel, no further views. Of course, a full 3D reconstruction is impossible in this case. But how could additional information help?

We restrict our choice of pictures to portraits of humans (though, Figs. 7 and 8 try out something different). We know, the image has a foreground where typically our human is pictured, and background that we would like to segment out and blur. We detect human skin colour for initialisation and engage further tricks—including user annotations—we detail below to find the watershed between foreground and background.

The central contribution of this work is the way how we combine skin detection, user annotations, and edge-preserving filters to obtain bluring masks, the coarse depth maps from a single image.

The next section handles related work, Section 3 presents our method, Section 4 shows the results, Section 5 presents the discussion, Section 6 concludes.

2 RELATED WORK

One of the first approaches for simulating bokeh effect were Potmesil and Chakravarty [PC81]; Cook [Coo86]. Most typical simulations of camera background blur

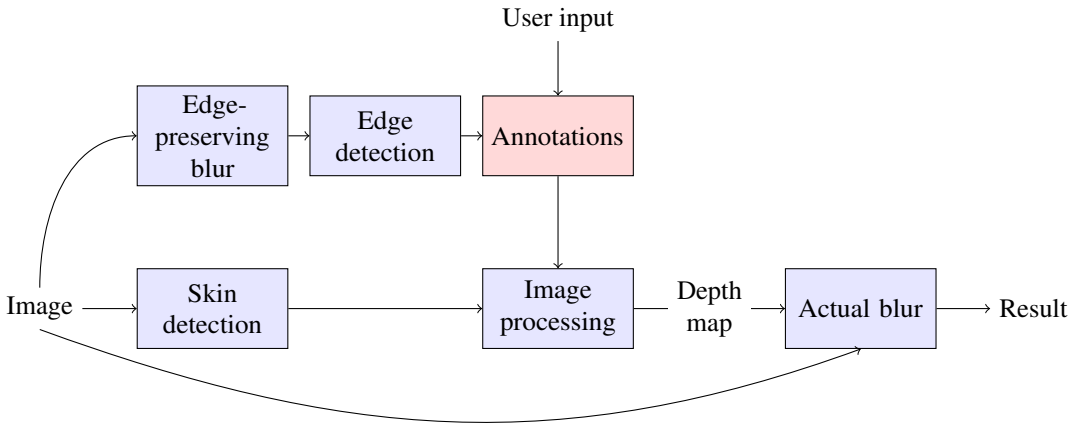


Figure 2: An overview of our approach. Everything that has skin colour is detected as foreground, then we add everything else where the user input matches on an image blurred in an edge-preserving manner. The different results are combined to a single mask. The mask and original input image are the input for bokeh simulation.

base on a full-fledged 3D scene, some of more recent methods are Wu et al. [Wu+12]; Moersch and Hamilton [MH14]. Yu [Yu04]; Liu and Rokne [LR12]; McIntosh, Riecke, and DiPaola [MRD12] discuss bokeh effect as a post-processing technique in rendering. This is different from our approach.

Nasse [Nas10] provides a nice technical overview of the bokeh effect. Sivokon and Thorpe [ST14] are concerned with bokeh effects in aspheric lenses.

Yan, Tien, and Wu [YTW09] are most similar to our approach, as they are concerned not only with bokeh computation, but also with foreground–background segmentation. They use a technique called "lazy snapping" [Li+04], we discuss the differences to our approach in Section 5.4.

A lot of research focuses on how to compute a realistic bokeh effect, given an image and its depth map, (see, e.g., [BFSC04]) It is in fact wrong to use a Gaussian blur (like [GK07] do) as the resulting image is too soft.

Lanman, Raskar, and Taubin [LRT08] capture the characteristics of bokeh and vignetting using a regular calibration pattern and then apply these data to further images. We rely on McGraw [McG14] in the actual bokeh *computation* from input data and estimated depth maps, which is a much more synthetic method as detailed below. This work actually focuses on obtaining the mask, "what to blur" from a single 2D image.

Bae and Durand [BD07] estimate an existing de-focus effect on images made with small sensors and amplify it to simulate larger sensors. This includes both the estimation of the depth map and the generation of a shallow depth-of-field image. Motivation of this work is very similar to ours, but the method is completely different. They estimate existing small defocus effects from the image and then amplify them using Gaussian blur.

Notably, Zhu et al. [Zhu+13] do the reverse of our approach. We estimate with some assumptions about the images and further inputs the foreground–background segmentation to compute then the depth-of-field effect. Zhu et al. estimate the foreground–background segmentation from shallow depth-of-field images. Works like Zhang and Cham [ZC12] concentrate on "refocusing," i.e., on detecting unsharp areas in a picture and on making the unsharp areas more sharp.

Saxena, Chung, and Ng [SCN07] present a supervised learning approach to the depth map estimation. This is different from our method. Saxena, Chung, and Ng divide the visual clues in the image into relative and absolute depth clues—evidences for difference of depth between the patches or for an "actual" depth. They use then a probabilistic model to integrate the clues to a unified depth image. This work does not focus on the computation of the shallow depth-of-field image. Eigen, Puhrsch, and Fergus [EPF14] use deep learning technique. A sophisticated neural network is trained on existing RGB+D datasets and evaluated on a set of other images from the same datasets. This is radically different from our approach. Aside from the presence of humans in the picture we make no further assumptions and utilize no previously computed knowledge. We have to use some auxiliary user input though. Eigen, Puhrsch, and Fergus [EPF14] also do not focus on the generation of shallow depth-of-field image.

3 METHOD

We chain multiple methods. First, the foreground mask expands to everything in the input image that has a skin colour. This way, we identify hands and other body parts showing skin. We expand the selection by selecting further pixels of the similar colour in the vicinity of already selected ones—we need to select all the skin, not just some especially good illuminated parts.

However, all this does not help with selection of clothes, as it can be of any colour or shape, a further problem is hair. For this sake we have allowed user input for the annotations of definitely foreground and definitely background areas. An attempt to expand the annotation (à la "magic brush" selection in photo-editing software) based on the actual input image would result in too small "cells" on some occasions and hence too much hysteresis—think: canny edge detection. For this reason we apply an edge preserving blur to the image used as input for "magic brush." This ensures higher-quality depth maps, separating the foreground (actual subject) and background. Given the depth map and initial input image, we apply the method of McGraw [McG14] to obtain the actual blurred image.

The "cells" we have mentioned above are actually regions with higher frequency than elsewhere in the image, that is: regions where edge detection would find a lot of edges. We futher discuss this issue in Section 5.3. An overview of our pipeline is in Figure 2.

3.1 Parts of our pipeline

Filtering approaches increase the edge awareness of our estimation. We use egde-preserving filtering [BYA15] as a part of our pipeline. Skin detection [EMH15] was part of our pipeline (see also [Bra98]). The depth maps were also processed with standard methods like erosion and dilation.

3.2 Neighbourhood detection

To detect similar-coloured pixels in the vicinity of pixels already present in the mask, we used the von Neumann neighbourhood (i. e., 4-connected). We used HSV colour space, the folklore solution for human skin detection. A naive implementation evidenced hysteresis: a pixel is deselected as it is deemed as background, but it is selected again because it has a similar colour as foreground. To amend this problem, we utilised canny edge detection on the image after edge-preserving blur. This reduces the number of falsely detected small edges. Now, in the von Neumann neighbourhood computation we check additionally if a pixel or its neighbours are on the edge. It is the case, we exclude these pixels from further processing.

3.3 The pipeline executed (Fig. 3)

Figure 3 demonstrates the processing steps on an example image (a). Fig. (b) shows the result of edge-preserving blur, the edge detection applied to it yields (d). Some parts of the image are already selected via skin detection (c). Basing on edges and user input, a full shape can be selected (e). We do not limit our approach to a single shape and to foreground only, as (f) shows. These intermediate results are then processed with erosion and dilation image filters, yielding (g). This final depth map is then applied to the input image (a) using the method of McGraw [McG14]. The final result is in (h).

4 RESULTS

4.1 Selfies

Our method works best on selfie-like images. Such images typically feature relatively large subject heads, further selfies are mostly captured on a mobile phone, thus they have a large depth-of-field. This fact makes them very suitable for an artistic bokeh simulation that is impossible to achieve with hardware settings in this case.

The input and reference images in Figure 1 were shot on a Canon 6D full-frame camera at 200 mm focal distance. To mimic the large depth-of-field of lesser cameras, the input image was captured at $f/32$, the reference image was captured at $f/4$ to showcase the real bokeh effect. The images were produced with Canon EF 70–200 mm $f/4$L lens. Our method works fine also when the head is relatively smaller in the whole picture (Fig. 4).

Featuring more than one person in a photograph is not a problem for our method, as Fig. 5 shows.

4.2 Multiple depths

Our depth maps facilitate not only a foreground–background segmentation, as showcased in Figs. 3, 6, and 7.

The input for Figure 6 was captured on a mobile phone and because of small sensor size it features a greater depth of field. Porting out application to mobile phones might be a promising way of using it. Fig. 7 also features multiple depth levels, we discuss it below.

5 DISCUSSION

We discuss following issues: how our method performs on non-human subjects of a photograph (Sec. 5.1), the issues with thin locks of hair (Sec. 5.2), we give more details on the cases when edge detection does not perform well (Sec. 5.3). Then we compare our method to "lazy snapping" (Sec. 5.4) and the result of our method to a real photograph with bokeh effect (Sec. 5.5).

5.1 Non-humans

We applied our method to Figs. 7 and 8. Naturally, no skin detection was possible here. The masks were created with user annotations on images after edge-preserving blur with canny edge detection as separator for different kinds of objects.

Note that in both examples, in case of the real shallow depth of field image, the table surface (Fig. 7) or soil (Fig. 8) would feature an area that is in-focus, as the focal plane crosses the table top or the ground. This is not the case in our images, as only the relevant objects were selected as foreground. Of course, it would be easy to simulate this realistic bokeh effect using a simple further processing of the depth map.

(a) Input image (b) Result of edge-preserving blur (c) Skin detection (d) Canny edges

(e) Depth map, an intermediate state (f) Adding a further level to the depth map, an intermediate state (g) Final depth map (h) Final result

Figure 3: Results of various intermediate steps of our pipeline. Input image (a) was captured at 27 mm full-frame equivalent at $f/2.8$ on a compact camera with crop factor 5.5. The binary foreground–background segmentation mask is in Fig. (g), final result with bokeh effect applied is in (h).

(a) Input image (b) Mask (c) Result

Figure 4: Filtering an image with head and shoulders. Input image (a) was captured using 57 mm full-frame equivalent lens at $f/4.5$ with crop factor 1.5.

(a) Input image (b) Mask (c) Result

Figure 5: Two persons in a photograph. Input image was captured at 43 mm focal distance equivalent on a full-frame, $f/5.6$, crop factor 1.5.

5.2 Hair

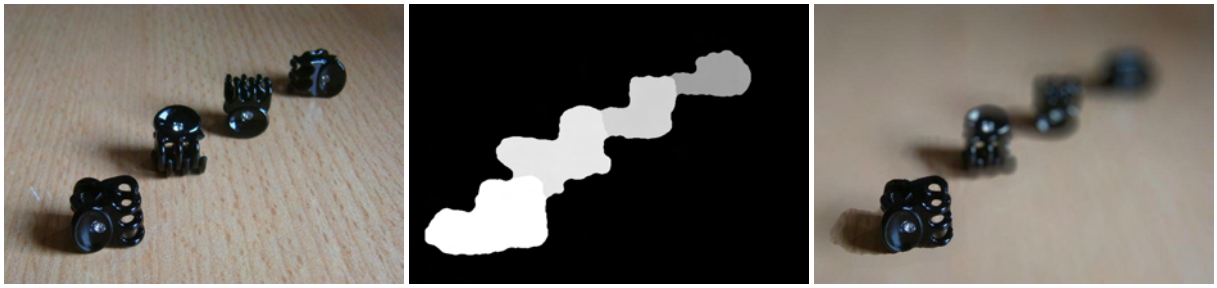
Thin flocks of hair cannot be easily detected, esp. on a nosily background. Automatic or annotation-based selection of such hair parts features a larger problem. Naturally, anything not present in the foreground selection enjoys background treatment during the actual bokeh simulation. One of most prominent visuals for such a side effect is Figure 9, even though some other our examples also showcase this issue.

5.3 Obstacles for edge detection

We use canny edge detection after an edge-preserving blur to separate "meaningful" edges from nonsense ones. This is basically the object segmentation that determines the boundaries of "cells" on which user annotations act. If an image features a lot of contrasts that survive the blur per Badri, Yahia, and Aboutajdine [BYA15], the user would require to perform more interactions than desired, as the intermediate result features too many

(a) Input image (b) Mask (c) Result

Figure 6: Showcasing more than a foreground and background separation. Input image captured on a mobile phone. The big plant on the left has a further depth level assigned.



(a) Input image (b) Mask (c) Result

Figure 7: Showcasing more than a foreground and background separation. This image has no humans on it. Input image (a) was captured at 27 mm full-frame equivalent at $f/2.8$ on a compact camera with crop factor 5.5.

"cells." Figure 10 illustrates this issue. Of course, a fine-tuning of edge-preserving blur parameters would alleviate this problem. However, we did not want to give our user any knobs and handles besides the quite intuitive input method for the "cell" selection, i.e., the annotations as such.

5.4 Comparison to lazy snapping

Yan, Tien, and Wu [YTW09] use lazy snapping [Li+04] and face detection for the segmentation. They typically produce gradients in their depth maps, to alleviate the issue we mentioned above in Section 5.1.

Lazy snapping uses coarse user annotations, graph cut, and fine-grain user editing on the resulting boundaries. In a contrast, we apply skin detection and edge detection on images blurred in an edge-preserving manner. The cells after edge detection are then subject to user annotations. We do not allow fine-grain editing of boundaries and thus drastically reduce the amount of user input, we are basically satisfied with coarse user annotations.

5.5 Comparison to real bokeh

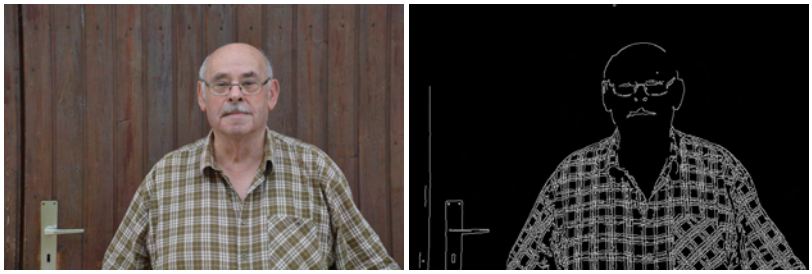
Compare images in the middle (our approach) and on the right hand side (ground truth) of Figure 1. We see a sharper edge in the hair, similarly to the issue discussed above. There is also a strange halo effect around the collar of the shirt. A further refinement and processing of the depth map data could help. Aside from these issues, the bokeh effect itself is represented quite faithfully. In an interesting manner, our synthetic image appears to be more focusing on the subject than the ground truth image. A possible reason is: the whole subject in our version is sharp. The ground truth version focuses on the eyes, but parts of the subject are already unsharp due to a too shallow depth-of-field: see shirt collar or the hair on the left. As our version is based on an image with a large depth-of-field (Fig. 1, left), it does not have these issues.

(a) Input image (b) Mask (c) Result

Figure 8: Applying our method to a photograph of a dog. By definition, no skin detection was possible. Captured on a mobile phone.

(a) Input image (b) Mask (c) Result

Figure 9: Limitation of our method: hair. Notice how some hair locks are missing in the mask and are blurred away. Captured at 69 mm full-frame equivalent at $f/4.8$ with crop factor 1.5.



(a) Input image (b) Canny edges

Figure 10: Limitation of our method: obstacles for edge detection. Input image (a) was captured at 82 mm full-frame equivalent at $f/6.3$ with crop factor 1.5. Note how the plaid shirt forms separate cells after canny edge detection (b), necessitating a larger annotation.

6 CONCLUSIONS

We have combined skin detection with user annotations to facilitate a coarse depth map generation from a single 2D image without additional modalities. The user input was processed on an extra layer after edge-aware blurring. In other words, we have enabled foreground–background separation through image processing and computer vision techniques and minimal user input. The resulting depth maps were then subsequently used to process the input image with a simulation of out-of-focus lens blur. Combined, we create a well-known lens effect ("bokeh") from single-image 2D portraits.

Future work

A mobile phone-based application might be of an interest, considering the selfie boom. Some UI tweaks like a fast preview loop after each user input and general performance improvements might be helpful in this case.

Face detection could be useful in general and for better handling of hair—we would use different parameters in the pipeline around the head, i.e., for hair, than everywhere else. Correct hair selection is probably the best area to further improve our work.

Further, our application benefits from any improvements in skin detection, edge-preserving blur, or bokeh simulation.

7 ACKNOWLEDGEMENTS

We would like to thank the photographers R. Friedrich, J. Kollmer, and K. Wölfel. Both the photographers and the models agreed that their pictures may be used, processed, and copied for free.

We thank T. McGraw, E. S. L. Gastal, M. M. Oliveira, H. Badri, H. Yahia, and D. Aboutajdine for being able to use their code.

REFERENCES

[BD07] S. Bae and F. Durand. Defocus magnification. *Comput. Graph. Forum*, 26(3):571–579, 2007.

[BFSC04] M. Bertalmio, P. Fort, and D. Sanchez-Crespo. Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In *3D data processing, visualization and transmission*, 2004, pages 767–773.

[Bra98] G. R. Bradski. Conputer vision face tracking for use in a perceptual user interface. *Intel technology journal*, 1998.

[BYA15] H. Badri, H. Yahia, and D. Aboutajdine. Fast edge-aware processing via first order proximal approximation. *IEEE T. Vis. Comput. Gr.*, 21(6):743–755, 2015.

[Coo86] R. L. Cook. Stochastic sampling in computer graphics. *ACM T. Graphic.*, 5(1):51–72, 1986.

[EMH15] A. Elgammal, C. Muang, and D. Hu. Skin detection. In, *Encyclopedia of Biometrics*, pages 1407–1414. Springer, 2015.

[EPF14] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In, *Adv. Neur. In.* Volume 27, pages 2366–2374. Curran, 2014.

[GK07] J. Göransson and A. Karlsson. Practical post-process depth of field. *GPU Gems*, 3:583–606, 2007.

[Li+04] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM T. Graphic.*, 23(3):303–308, 2004.

[LR12] X. Liu and J. Rokne. Bokeh rendering with a physical lens. In *PG '12 Short proc.* EG, 2012. ISBN: 978-3-905673-94-4.

[LRT08] D. Lanman, R. Raskar, and G. Taubin. Modeling and synthesis of aperture effects in cameras. In. In COMPAESTH '08. EG, 2008. ISBN: 978-3-905674-08-8.

[McG14] T. McGraw. Fast bokeh effects using low-rank linear filters. *Visual Comput.*, 31(5):601–611, 2014.

[MH14] J. Moersch and H. J. Hamilton. Variable-sized, circular bokeh depth of field effects. In *Graphics Interface '14*. CIPS, 2014, pages 103–107.

[MRD12] L. McIntosh, B. E. Riecke, and S. DiPaola. Efficiently simulating the bokeh of polygonal apertures in a post-process depth of field shader. *Comput. Graph. Forum*, 31(6):1810–1822, 2012.

[Nas10] H. H. Nasse. Depth of field and bokeh. *Carl Zeiss camera lens division report*, 2010.

[PC81] M. Potmesil and I. Chakravarty. A lens and aperture camera model for synthetic image generation. *SIGGRAPH Comput. Graph.*, 15(3):297–305, 1981.

[SCN07] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *Int. J. Comput. Vision*, 76(1):53–69, 2007.

[ST14] V. P. Sivokon and M. D. Thorpe. Theory of bokeh image structure in camera lenses with an aspheric surface. *Opt. Eng.*, 53(6):065103, 2014.

[Wu+12] J. Wu, C. Zheng, X. Hu, and F. Xu. Rendering realistic spectral bokeh due to lens stops and aberrations. *Visual Comput.*, 29(1):41–52, 2012.

[YTW09] C.-Y. Yan, M.-C. Tien, and J.-L. Wu. Interactive background blurring. In. In MM '09. ACM, 2009, pages 817–820.

[Yu04] T.-T. Yu. Depth of field implementation with OpenGL. *J. comput. sci. coll.*, 20(1):136–146, 2004. ISSN: 1937-4771.

[ZC12] W. Zhang and W.-K. Cham. Single-image refocusing and defocusing. *IEEE T. Image Process.*, 21(2):873–882, 2012.

[Zhu+13] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar. Estimating spatially varying defocus blur from a single image. *IEEE T. Image Process.*, 22(12):4879–4891, 2013.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

3D Mesh Simplification for Deformable Human Body Mesh Using Deformation Saliency

Tianhao ZHAO
R301, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
thzhao@ee.cuhk.edu.hk

King Ngi NGAN
R304, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
knngan@ee.cuhk.edu.hk

Songnan LI
R301, SHB, CUHK
Shatin, N.T.
999077, Hong Kong
snli@ee.cuhk.edu.hk

ABSTRACT

3D mesh of human body is the foundation of many hot research topics, such as 3D body pose tracking. In this topic, the deformation of the human body mesh has to be taken into account because of various poses of the human body. Considering the time cost of the body deformation, however, it's impractical to adopt a high resolution body mesh generated from scanning systems for the real-time tracking. Mesh simplification is a solution to reduce the size of body meshes and accelerate the deformation process.
In this paper, we propose a mesh simplification algorithm using deformation saliency for such deformable human body meshes. This algorithm is based on quadric edge contraction. The deformation saliency is computed from a set of meshes with various poses. With this saliency, our algorithm can simplify the 3D mesh non-uniformly. Experiment shows that using our algorithm can improve the accuracy of body pose simulation in the simplified resolution compared to using classical quadric edge contraction methods.

Keywords

Mesh simplification, Deformable human body mesh, Deformation saliency.

1 INTRODUCTION

Nowadays, 3D human body tracking is a hot research topic [Bog15, Baa13, Wei12, Gan10]. In this topic, 3D body mesh is a basic structure. Generally, the input of 3D body tracking is a sequence of depth images captured by depth cameras. A body mesh in an initial pose is selected as a template mesh. The template mesh is deformed to different poses according to the real-time input frames. The body mesh is defined as the combination of a point cloud and a set of triangulated faces. The faces cover all the points and there is no overlay between any two faces. Typically the 3D mesh is obtained from a multiple depth-camera scanning system or a laser scanning system. The mesh generated by these scanning systems is in very high resolution, containing tens of thousands of vertices and faces, such as the CAESAR dataset. Nevertheless, 3D human body tracking is required to be real-time, so the body mesh in such high resolution is not suitable for the real-time tracking. Reducing the mesh resolution, i.e., the number of vertices and faces of the mesh is necessary.

In this paper, we focus on the mesh simplification for such deformable human body object. We adopt the deformable human body model proposed by [Ang05]. The body meshes of SCAPE model are from the CAESAR dataset. The mesh standing in the "A" pose (shown in Fig.1) is selected as the template mesh and other meshes contain various poses of the same person. All the meshes have the same number of vertices, which have been registered. In different meshes, the topologies of triangulated faces are also the same. The human body is partitioned into 16 parts, containing head, chest, torso, arms, hands, legs and feet. The motion model of SCAPE describes two kinds of deformations of human bodies: the rigid transformation and non-rigid deformation. The rigid transformation matrices show the global transformation of an integral body part. For all the vertices in the same body part, their rigid transformation matrices of a certain pose are the same. The non-rigid deformation matrices indicate the change of muscle, skin and ligament in different poses. These matrices are unique for each face. Apparently, there are much more non-rigid deformations at joints or muscular regions than other body regions. Therefore, in simplification more vertices in these special regions should be preserved to simulate the non-rigid deformation more accurately.

However, there is a drawback of the traditional mesh simplification algorithms when they are applied to such deformable meshes: they always simplify a 3D mesh

uniformly or based on geometric features, which means that more vertices will be kept in some geometry-salient regions, like fingers and face of the human. This may lead to very few vertices remaining in the highly non-rigid deforming regions. Hence, our purpose is to distinguish the highly non-rigid deforming regions, and keep more vertices in these regions when simplifying the body mesh.

Based on the quadric edge contraction (Qslim) [Gar97], we developed a mesh simplification algorithm guided by deformation saliency. In our algorithm, we compute the deformation saliency for every vertex in the template mesh from the SCAPE pose database. Besides the deformation saliency, we also introduce a distance balancing item to avoid the over-contraction. Our method can simplify meshes region-specifically according to the non-rigid deformation levels. Experimental result shows that using our simplification method, the deformed result is more accurate than that using the Qslim.

This paper is organized as following. Section 2 introduces the related work. Section 3 discusses our methodology. Section 4 analyzes experimental results and compares the proposed method with the prior studies. Conclusion is drawn in Section 5.

2 RELATED WORK

Mesh simplification aims to reduce the number of vertices of a 3D mesh, meanwhile preserving the shape of the mesh as accurately as possible. To this end, the Quadric Edge Collapse Decimation (Qslim) was proposed by [Gar97]. In their paper, each vertex of the mesh was associated with a quadric error matrix, which was defined as the squared distances from this vertex to the planes of its adjacent triangulated faces. Every edge had a contraction cost based on its potential contraction-target vertex and the quadric error matrices of its two endpoints. Iteratively the edges with the minimum contraction costs were contracted to the optimal target vertex which could minimize the contraction costs. The contraction costs for all related vertices would be updated in each iteration.

One trend to improve the mesh simplification method is to reduce the running time. Using GPU is a common way to speed up the computation. [Sho13] proposed a CPU-GPU combined algorithm, which has the lowest computational cost compared to all the other methods just running on CPU. Some methods like [Cam13] considered both accuracy and simplicity to obtain the optimal simplification result. Another multilevel refinement method was proposed in [Mor14], which combined a Laplacian flow to the high resolution mesh. This method can locate the most appropriate regions to be contracted in different refinement levels, which took into account both accuracy and speed.

Another trend to improve the simplification is to keep more details. Mesh features are used widely in the related work. More vertices in the regions with prominent features will be preserved. For example, mesh curvature is a common feature used in the prior studies [All03, Lee05, Wan11, Yao15] . [All03] introduced curvature directions to represent the intrinsic anisotropy of the mesh geometry. In [Lee05], mesh feature was defined as a center-surround operator on Gaussian-weighted mean curvatures. In [Wan11], by measuring the curvature, the authors proposed a method to perform coarse simplification in flat regions and fine simplification near creases and corners respectively. [Yao15] adopted the discrete curvature to modify the Qslim method.

Besides curvature, there are many other saliencies used for mesh simplification. [Tol08] introduced acceleration and deceleration of vertices as saliency for simplification of dynamic meshes. [Zha12] identified visually important regions by points sampling method to keep the mesh saliency. [Pey14] reduced the number of vertices by Possion disk sampling based on features such as sharp edges or corners, and re-meshed vertices along the detected feature lines. In [Pel14], Pellerin et al. applied Centroidal Voronoi optimization to simplify the mesh and merged features for the mesh with complex contacts. A B-rep feature-based model was proposed in [Kim14]. [Ng14] developed a method called half-edge collapsed scheme. They identified valid decimated edges by the length of the edges and the difference between every two adjacent faces. Another distinctive method was proposed in [Van15], which adopted outgoing radiance functions of the mesh surface as the mesh feature. In [Ďur15], shape diameter function was adopted as mesh saliency to extract skeleton, which can also be used in mesh simplification.

3 METHODOLOGY

Our purpose is to simplify the human body mesh while preserving more vertices at joints and other regions with prominent non-rigid deformations. Given a set of meshes M of various poses and a template mesh T of the SCAPE data set, we compute the deformation saliency along with a balancing weight for each vertex. Then we iteratively contract a valid edge on the template mesh to generate a new mesh T' in lower resolution. The indices of the preserved vertices are recorded. For the other meshes, the vertices with the same indices will be kept, so that the topologies of the triangulated faces of these meshes are the same as that of the simplified template mesh T'.

3.1 Deformation saliency

We define the deformation saliency as the Euclidean distance of the corresponding vertices between the

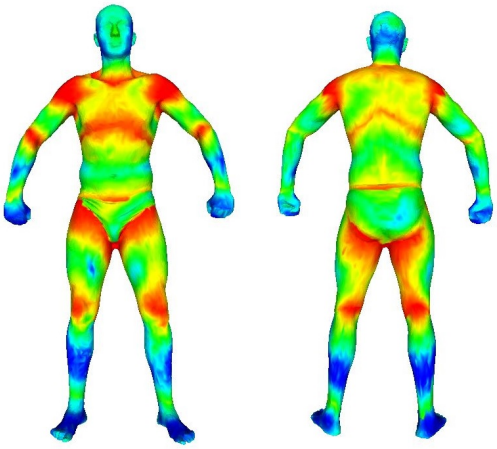


Figure 1: The heat maps of saliency values on the template mesh. Red means the highest value, and blue means the lowest value

rigidly transformed template mesh and the mesh in the target pose, which is also described by the non-rigid deformation in the SCAPE model. First we estimate the rigid transformation between the template mesh T and each mesh M_X of pose X in the pose dataset, which means to calculate the integral rotation matrix for each body part between mesh T and M_X in the global coordinates. In the SCAPE database, vertices have been registered across different meshes and partitioned into 16 body parts. For each body part k, the rotation matrix and translation vector are denoted as R_k and t_k, respectively. To estimate R_k and t_k, we minimize the following energy function:

$$E_{transform} = \underset{R_k,t_k}{\operatorname{argmin}} \sum_{v_i^T, v_i^{M_X} \in \text{part}(k)} \|R_k v_i^T + t_k - v_i^{M_X}\|_2^2 \tag{1}$$

In this least square function, v_i^T is the ith vertex in the mesh T and $v_i^{M_X}$ is the ith vertex in the mesh M_X.

By vectorizing R_k and t_k, Eq. (1) can be turned into a system of linear equations, which can be solved analytically. With the rigid transformation, we can transform each body part of the template mesh via this equation:

$$v_i^{T_X} = R_k v_i^T + t_k, v_i \in \text{part}(k) \tag{2}$$

where T_X is the rigidly transformed template mesh corresponding to the mesh M_X.

After generating the rigidly transformed meshes T_X in all poses X, we calculate the Euclidean distance error between every vertex of T_X and its corresponding vertex of M_X. For each vertex, we add up its distance errors calculated from all the poses X. Then we normalize the total error and take it as the deformation saliency for each vertex. Fig.1 shows the saliency values around the human body.

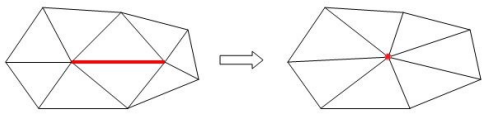


Figure 2: The red edge denotes the edge with the minimum deformation weight and contraction cost; it is contracted to a new vertex (the red point) which can minimize the contraction cost; other related vertices are re-connected to the new vertex.

The vertex with high value of deformation saliency means there is prominent non-rigid deformation, so in the simplification its possibility to be preserved should be larger than other vertices. As the Qslim algorithm illustrated [Gar97], every vertex holds a quadric matrix Q, which represents the entire set of planes adjacent to this vertex. For an edge (v_1, v_2), the contraction cost associated to its potential target vertex v_t is defined as:

$$E_{contract} = v_t^T (Q_1 + Q_2) v_t \tag{3}$$

In this equation, Q_1 and Q_2 are the quadric matrices of v_1 and v_2, respectively. v_t is the optimal target vertex which can minimize the contraction cost $E_{contract}$. The contraction cost indicates the sum of squared distances from v_t to the plane set represented by Q_1 and Q_2.

Using a couple of weights w_1, w_2 to represent the deformation saliencies of v_1, v_2, we update the edge contraction cost as:

$$E_{contract} = \frac{w_1 + w_2}{2} \cdot v_t^T (Q_1 + Q_2) v_t \tag{4}$$

The deformation weight of an edge is defined as the average weight of its two attached endpoints. However, an extreme result is that too many edges may be contracted in the regions where vertices have low deformation weights. In this case, very long edges may be generated; meanwhile few edges are contracted in the regions where vertices have high deformation weights. To solve this dilemma, we introduce a balancing weight for each edge, which is equal to the length of the edge (v_1, v_2). The effect of this balancing weight is to reduce the contracting priority of long edges. By denoting this balancing weight as d, the edge contraction cost is rewritten as:

$$E_{contract} = \frac{d(w_1 + w_2)}{2} \cdot v_t^T (Q_1 + Q_2) v_t \tag{5}$$

Iteratively, the edge with the least contraction cost is contracted, generating the simplified template mesh T'.

Fig.2 illustrates edge contraction in a single iteration, and Fig.3 compares the simplified results with and without the balancing item.

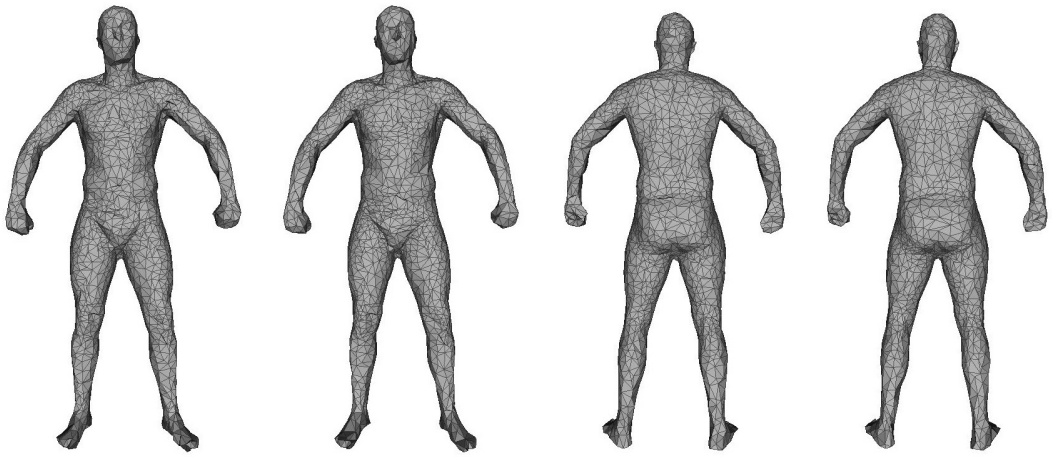


Figure 3: From left to right: front side of the simplified template mesh with and without balancing item; back side of the simplified template mesh with and without balancing item. Edges of the meshes are shown explicitly. With the balancing item, the variance of edge length is 1.53e-04; without it, the variance of edge length is 3.08e-02. This proves that the balancing item can prevent too long edges occurring effectively.

3.2 Greedy least distance mapping

According to the SCAPE, to build the body deformation model, vertex partitions and face topologies of all the meshes should be the same, and all the meshes should be registered. So we use a greedy least distance mapping from T' to T to keep vertices registered across different simplified meshes.

Initially, we define the simplified candidates as all vertices of T' and the original candidates as all vertices of T. Iteratively, between the two candidates we find the mapping vertex pair with the least Euclidean distance, record the vertex index of this pair, and take them out from the candidates. All vertices of T' are mapped to distinctive vertices of T. With the mapping record, we can simplify other meshes via preserving the vertices with the same indices as in the record. As a consequence, all simplified meshes still contain the registered vertices and the same body partition.

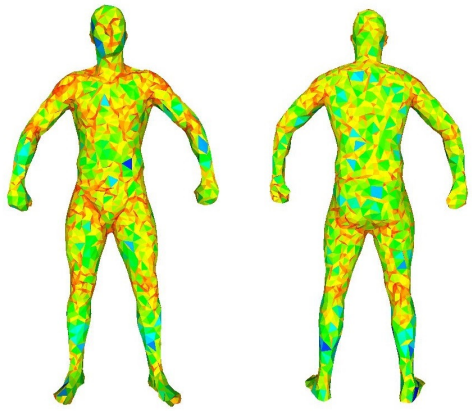


Figure 4: The area heat maps of the simplified template mesh using our method (2500 vertices, 5000 faces).

4 EXPERIMENTS

We conducted the experiments on a subset of the SCAPE database. We chose 70 body meshes in different poses of the same person as the pose dataset. Each of them has 12500 vertices and 25000 triangulated faces. In the 70 meshes, the first mesh standing in the "A" pose was selected as the template mesh. Based on the pose set, deformation saliency was computed. With our method and the compared methods, the template mesh and other meshes were simplified. The number of faces and vertices were reduced from 25000 to 5000 and from 12500 to 2500, respectively. Then the template mesh was deformed to other poses. The errors between the deformed template mesh and the mesh of the target pose were computed to evaluate the proposed methods.

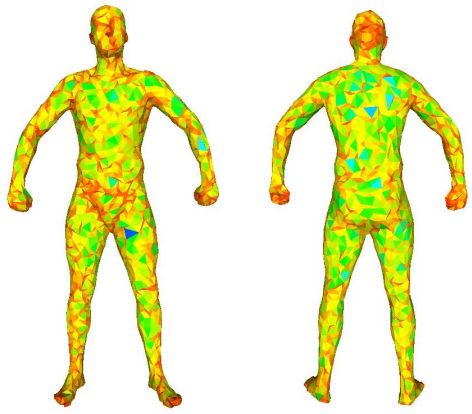


Figure 5: The area heat maps of the simplified template mesh using Qslim (2500 vertices, 5000 faces).

4.1 Area heat map of the simplified result

The visualized comparison between the results of the proposed method and the Qslim method is drawn in Fig.4 and Fig.5, which are referred to as the heat maps of the area of the triangulated faces.

In both figures, the faces are colorized according to the size of their area: smaller faces have warmer color, while larger faces have cooler color. That is to say, red regions have the highest vertex density, and blue regions have the lowest vertex density. Yellow and green regions have median vertex density.

By observing the two heat maps, we can find that vertex densities are quite different in the same region across the results produced by different methods. We know the head, hands, and feet typically have more geometric details. Therefore, the Qslim method preserves more vertices in these regions. On the contrary, it is clear that we preserve more vertices at shoulders, knees, elbows, even in chest and thighs; meanwhile we preserve fewer vertices in the regions like hands, head and feet. This means that we have more vertices to simulate the non-rigid deformation at joints and muscular regions.

4.2 Deformation errors

In this section, we measure the deformation errors between the deformed template mesh and the mesh of the target pose. Less deformation error means that the simplification method is more suitable for such deformable human body mesh to change the body pose. The detail of the body pose deformation can be referred to [Ang05]. Besides the 70 meshes simplified by our method and the Qslim, we also simplify 10 meshes using the method provided in the open-source CGAL library for comparison. The CGAL (http://www.cgal.org/) is a widely-used library of geometry algorithms. We introduce two criteria to measure the deformation errors:

1. Vertex-to-face error (v2f)

The vertex-to-face error measures the average squared distance between the deformed template mesh and the simplified mesh of the target pose, from a vertex to the closest face. It is also adopted by the Qslim method [Gar97]. The vertex-to-face error $E_i = (M_i, M_n)$ between the deformed template M_i and ground-truth M_n is defined as:

$$E_i = \frac{1}{|X_i| + |X_n|} \left(\sum_{v \in X_i} d_1^2(v, M_n) + \sum_{v \in X_n} d_1^2(v, M_i) \right) \quad (6)$$

where X_i, X_n are the point clouds in the mesh M_i and M_n respectively. The distance function $d_1(v,M) = \min_{p \in M} \|v - p\|$ is the minimum Euclidean distance from the vertex v to the closest face p on the mesh M. The measurement unit of this error is $meter^2$.

Method	v2f error	v2v error
Ours(70 poses)	1.08e-09	3.83e-05
Qslim(70 poses)	1.12e-09	4.14e-05
CGAL(10 poses)	1.10e-09	3.93e-05

Table 1: The average deformation errors of our method, Qslim and CGAL.

2. Vertex-to-vertex error (v2v)

The vertex-to-vertex error measures the average squared distance from a vertex of the deformed template mesh to the closest vertex of the simplified mesh of the target pose. This metric is adopted in [Bog15]. Based on the previous notations, the vertex-to-vertex error $E_i = (M_i, M_n)$ is defined as:

$$E_i = \frac{1}{|X_i|} \sum_{v \in X_i} d_2^2(v, M_n) \quad (7)$$

The distance function $d_2(v,M) = \min_{u \in M} \|v - u\|$ is the minimum Euclidean distance from the vertex v to the closest vertex u on the mesh M. The measurement unit is also $meter^2$.

The comparison results between our method and other methods are shown in Table 1. The average errors show that our method can reduce the deformation errors in both metrics. The reduction of the v2v error is more significant, which approaches about 8 percent compared to the Qslim. The results indicate that using our method, the simplified human mesh can be deformed more accurately than using the Qslim and CGAL.

Fig.6 shows some simplified results of different identities with the same pose, and Fig.7 shows the pose deformation results after simplification for several poses.

5 CONCLUSION

In this paper, we propose a mesh simplification method using deformation saliency for 3D human pose deforming. Based on SCAPE dataset, we compute the vertex distance error caused by non-rigid deformation for each vertex as the deformation saliency. In the template body mesh we contract the edges with the lowest saliency in prior. We also add a balancing weight to avoid generating too long edges caused by over-contraction. Our method shows better performance of body pose deformation compared to the Qslim and CGAL algorithms. Similarly, our method can be also applied to simplifying the meshes of other deformable objects for pose deformation.

6 REFERENCES

[Bog15] Bogo F, Black M J, Loper M, et al. Detailed Full-Body Reconstructions of Moving People from Monocular RGB-D Sequences. Proceedings of the IEEE International Conference on Computer Vision, pp.2300-2308, 2015.

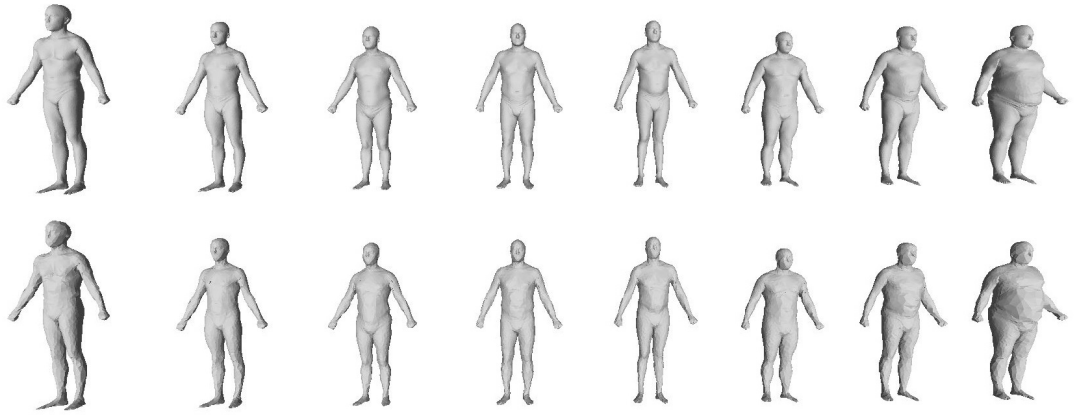


Figure 6: The first row shows several examples of the high resolution meshes from the SCAPE database (12500 vertices and 25000 faces); the second row shows our simplified meshes (2500 vertices and 5000 faces).

[Baa13] Baak A, Muller M, Bharaj G, et al. A data-driven approach for real-time full body pose reconstruction from a depth camera. Consumer Depth Cameras for Computer Vision, pp.71-98, 2013.

[Wei12] Wei X, Zhang P, Chai J. Accurate realtime full-body motion capture using a single depth camera. ACM Trans. Graph., 31(6), pp.188, 2012.

[Gan10] Ganapathi V, Plagemann C, Koller D, et al. Real time motion capture using a single time-of-flight camera. Computer Vision and Pattern Recognition, pp.755-762, 2010.

[Ang05] Anguelov D, Srinivasan P, Koller D, et al. SCAPE: shape completion and animation of people. ACM Trans. Graph., 24(3), pp.408-416, 2005.

[Gar97] Garland M, Heckbert P S. Surface simplification using quadric error metrics. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp.209-216, 1997.

[Sho13] Shontz S M, Nistor D M. CPU-GPU algorithms for triangular surface mesh simplification. Proceedings of the 21st international meshing roundtable, pp.475-492, 2013.

[Cam13] Campomanes-Alvarez B R, Cordon O, Damas S. Evolutionary multi-objective optimization for mesh simplification of 3D open models. 20(4), pp.375-390, 2013.

[Mor14] Morigi S, Rucci M. Multilevel mesh simplification. The Visual Computer, 30(5), pp.479-492, 2014.

[All03] Alliez P, Cohen-Steiner D, Devillers O, et al. Anisotropic polygonal remeshing. ACM Trans. Graph., 22(3), pp.485-493, 2003.

[Lee05] Lee C H, Varshney A, Jacobs D W. Mesh saliency. ACM Trans. Graph., 24(3), pp.659-666, 2005.

[Wan11] Wang J, Wang L, Li J, et al. A feature preserved mesh simplification algorithm. Journal of Engineering and Computer Innovations, 6, pp.98-105, 2011.

[Yao15] Yao L, Huang S, Xu H, et al. Quadratic Error Metric Mesh Simplification Algorithm Based on Discrete Curvature. Mathematical Problems in Engineering, 2015.

[Tol08] Tolgay A. Animated mesh simplification based on saliency metrics. bIlkent university, 2008.

[Pey14] Peyrot J L, Payan F, Antonini M. Aliasing-free simplification of surface meshes. International Conference on Image Processing, pp.4677-4681, 2014.

[Pel14] Pellerin J, Levy B, Caumon G, et al. Automatic surface remeshing of 3D structural models at specified resolution: A method based on Voronoi diagrams. Computers and Geosciences, 62, pp.103-116, 2014.

[Kim14] Kim B C, Mun D. Feature-based simplification of boundary representation models using sequential iterative volume decomposition. Computers and Graphics, 38, pp.97-107, 2014.

[Ng14] Ng K W, Low Z W. Simplification of 3D Triangular Mesh for Level of Detail Computation. Computer Graphics, Imaging and Visualization, pp.11-16, 2014.

[Van15] Vanhoey K, Sauvage B, Kraemer P, et al. Simplification of meshes with digitized radiance. The Visual Computer, 31(6-8), pp.1011-1021, 2015.

[Zha12] Zhao Y, Liu Y, Song R, et al. A saliency detection based method for 3d surface simplification. Acoustics, Speech and Signal Processing, pp.889-892, 2012.

[Ďur15] Ďurikovič R, Madaras M. Controllable Skeleton-Sheets Representation Via Shape Diameter Function. Mathematical Progress in Expressive Image Synthesis II, pp.79-90, 2015.

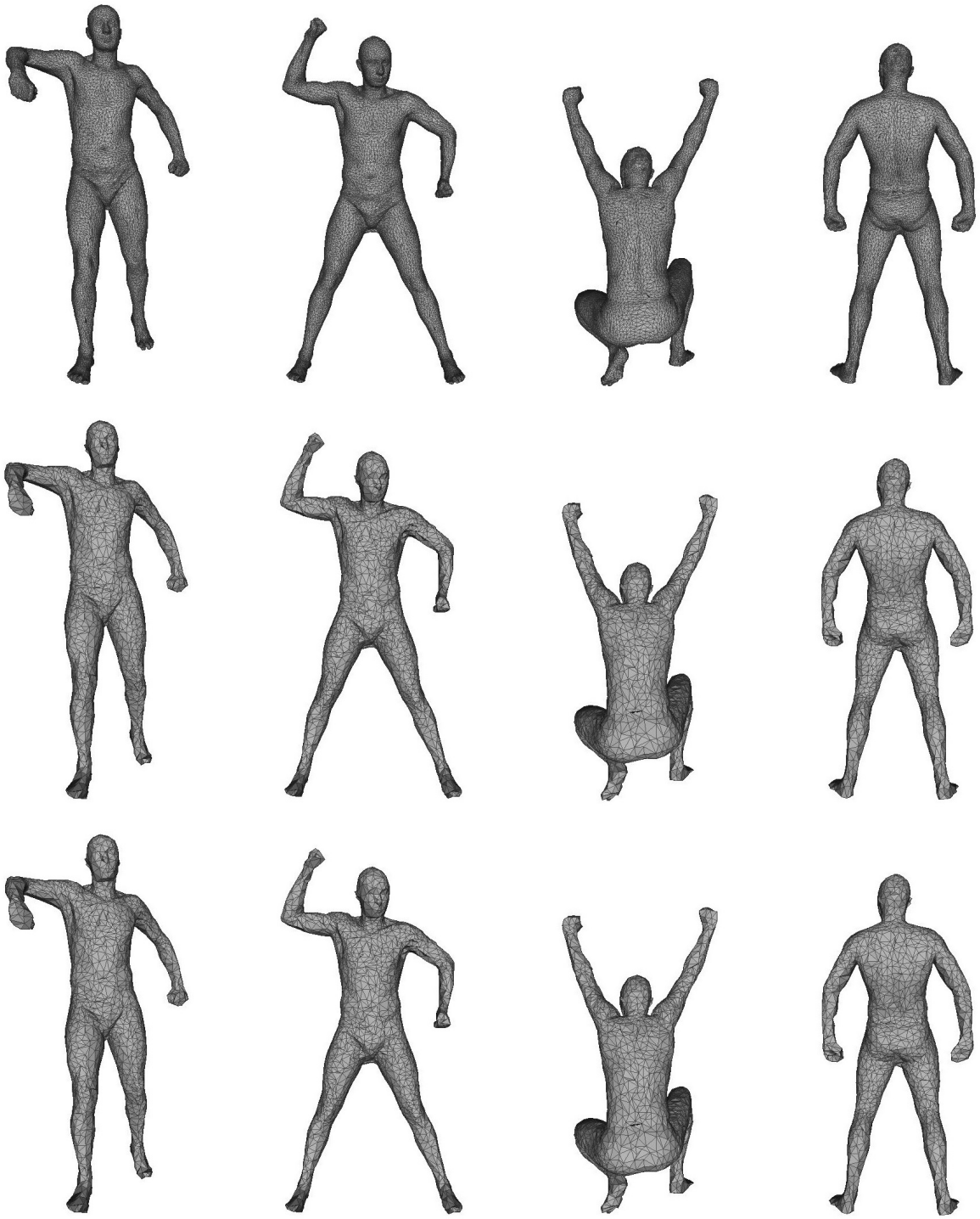


Figure 7: The first row shows some original meshes of different target poses; the second row shows the simplified results of these target poses; the third row shows the deformed results from the simplified template mesh to these target poses.

Registration of Deformable Objects using a Depth Camera

Rubén de Celis
Vicomtech-IK4 Foundation
Mikeletegi 57, SPAIN, 20009, Donostia
rdcelis@vicomtech.org

Nagore Barrena
Vicomtech-IK4 Foundation
Mikeletegi 57, SPAIN, 20009, Donostia
nbarrena@vicomtech.org

Jairo R. Sanchez
Vicomtech-IK4 Foundation
Mikeletegi 57, SPAIN, 20009, Donostia
jrsanchez@vicomtech.org

Ramón J. Ugarte
Vicomtech-IK4 Foundation
Mikeletegi 57, SPAIN, 20009, Donostia
rugarte@vicomtech.org

ABSTRACT

This paper describes a method for registration and tracking of deformable objects from points clouds taken from depth cameras. Our method uses a reference model of the object in order to detect rigid and deformed regions in the input cloud. It is based on the fact that deformed objects normally have areas that are not affected by the deformations. These parts are found iteratively allowing to register the object using a chain of rigid transformations. Deformed regions are detected as those that do not satisfy rigidity constrains. Results show that correspondences of points belonging to both rigid and deformed regions can be accurately established with the reference model even in cluttered scenes.

Keywords

3D Tracking, Deformable Object Recognition, Depth Camera

1 INTRODUCTION

The research presented in this paper is motivated by the need to detected deformations in elastic volumes for augmented reality applications and mechanical simulations. The field of computer vision already presents important advances in the tracking of non-rigid surfaces using monocular images [**?**]. These methods are usually based on geometric constraints applied to the analytical models of the objects to be detected. These models define the deformable objects as surfaces, and seek for results that are visually attractive. However, their physical behaviour involves properties such as elasticity, which affect their mechanical behaviour that cannot be well modelled with the cited techniques.

In the last years a great research effort has been done in the field of 3D reconstruction and object tracking thanks to the emergence of commodity depth cameras. They can give depth information for image pixels, usually using infrared technology. These kind of devices are particularly appropriate for the problem stated in this work as they allow to obtain a point cloud representation of the scene easily in real-time.

The basis of 3D object recognition involves finding a set of correspondences between a known reference object and the reconstructed scene. This problem is ususally solved under rigidity assumptions that allows registering the scene using an euclidean transformation. However, in the case of deformed objects the complexity of the problem increases as there is no a unique transformation that registers all the scene points with the reference.

In this work we pose the problem of deformable object registration as a recursive rigid registration problem. In our approach non deformed parts of the model are iteratively registered, representing the scene as a chain of rigid transformations. As a result points corresponding to deformed regions can be precisely detected and matched with the reference object in a straightforward manner. The main contribution of this method mainly relies on its simplicity which enables fast and robust implementations.

The paper is organized in four main sections: the "Related Work" section introduces the reader in the state of the art and develop tools, the "Method Overview" presents the execution pipeline of the method, the "Results" section is where are performed and discussed the data obtained, and "Conclusion and Future Work" shows the conclusion reached and open the following step in the investigation line.

2 RELATED WORK

The registration of 3D scenes is a well known problem that can be defined as the alignment of two different

point clouds representing the same scene. The emergence of depth cameras, such as Kinect, has spawned new interests in this line during the last years. One of the most representative works in this area was developed in [**?**] describing the techniques that are nowadays most used to register 3D objects. The main contribution of this research was a representation of objects as point clouds based on 3D feature histograms [**?**]. This representation describes the local geometry of object points relative to their neighbourhood that can be used to match point correspondences between different reconstructions. The work resulted in an open source library called Point Cloud Library (PCL) [**?**].

Most of the research works, like [**?**] and [**?**], assume rigid conditions for the objects to be scanned. Under this assumption point clouds can be aligned using a single euclidean transform. The most common registration techniques rely on the use of 3D keypoints detectors and descriptors in order to get correspondences that allow to find the transformation.

Some 3D detectors have been developed inspired by 2D image detectors like SIFT [**?**]. For now, a small set of detectors has been proposed specifically for 3D point clouds and range maps being ISS [**?**] and NARF [**?**] the most representatives.

Concerning descriptors, a commonly accepted taxonomy divides the descriptors in local, such as 3DSC [**?**], FPFH [**?**] or SHOT [**?**]; and global, as CVFH [**?**], ESF [**?**] or VFH [**?**]. Local descriptors are calculated for individual points being suitable for handling cluttered scenes and partially occluded objects. Global descriptors encode the object geometry, having higher invariance and being more descriptive. They are very suitable for the retrieval and classification of objects with poor geometric structure.

There are also very relevant advances in the registration of 3D scenes containing deformable elements. Works like [**?**], [**?**] or [**?**] illustrate cited procedure based on iterative minimization techniques. The input data is aligned with the reference model by minimizing an energy function that depends on various geometrical constraints. The main problem of these kind of methods is the existence of local minima in the objective function that cannot be always avoided. Method described in [**?**], simultaneously solves correspondences between points on source and reference clouds using an energy function that penalizes huge deformations and favours rigidity and consistency. The method implements a graph of nodes, whose nodes are chosen by uniform sampling, and each node have influence over the deformation of the nearby nodes. The computational cost is exponential with respect to the nodes and depends on the resolution used to generate the graph. The approach presented in [**?**], implements non-rigid reconstruction pipeline on the GPU and his approach include a custom

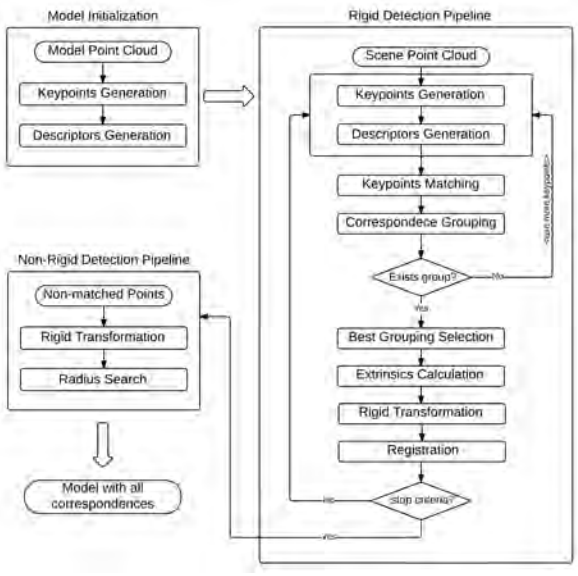


Figure 1: Detection Flow Diagram has 3 phases (Model Initialization, Rigid Detection Pipeline and Non-Rigid Pipeline).

RGB-D camera. The deformations between two scans are given by ARAP framework [**?**] that measures deformations existing between a pair of meshes. This type of registration is not useful when the goal is to detect deformations, because they perform deformations in the corresponding representation during the input data aggregation process.

Our method, unlike [**?**] or [**?**], is not based on iterative minimization frameworks. Instead we rely on simpler point correspondences that, besides simplicity, allow to avoid local minima as we can directly obtain the involved transformations once point correspondences are found.

3 METHOD OVERVIEW

The method proposed is designed to detect the deformations on the surface of the object. Given a reference model and the objects found in the scene, the method detects correspondences between the reference model and the model found in the scene, including undeformed and deformed regions. Figure 1 illustrates, the detection flow, divided in three phases. In the first phase, the reference model is initialized; in the second phase, the rigid regions considered as undeformed regions are detected; in the third phase, the non rigid regions considered as deformed regions are detected. The internal representation of the real world is based on point clouds, without edges.

In the initialization phase, the 3D keypoints of the model and their descriptors are computed from its point cloud representation. With the reference model initialized, the tracking is performed using a sequence of 3D

scans of the state of the scene as input. In the rigid detection phase, following the same procedure as in the initialization, the keypoints and descriptors are computed from the point cloud of the scene. Once the 3D keypoints and descriptors are computed, the keypoints are matched between the reference model and the scene, based on their descriptors information. Then all the correspondences are grouped in order to cluster the set of correspondences into instances that are present in the scene. An instance is defined as a subset of keypoints of the reference model matched with scene keypoints that satisfy a geometric consistency with the reference model. The best instance is used to calculate a rigid transformation to seek points corresponding to undeformed regions. With all the undeformed regions, the non-rigid detection phase is started with the non-matched points from the previous phase as input. Points of deformed regions are transformed with the best rigid transformation obtained in the previous phase. After applying the transformation, a radius search is executed to find for each non-matched point of the reference model the corresponding point in the scene.

The following subsections explain the phases in more detail.

3.1 Model Initialization

The reference model is represented as a point cloud. It can be loaded from a CAD or captured from a 3D scanner, provided that it is undeformed. The initialization process consists in the selection and computation of a set of keypoints and their descriptors from the cloud.

In order to obtain a good representation which enables a stable tracking, it is important to perform a proper selection of keypoints. There are well known detectors such as ISS and NARF which use the gradient of the surface around the vicinity to detect representative keypoints. Although there are good candidates to perform the matching of rigid surfaces, they are not appropriate to deformable models because the surface gradient is not invariant.

For this reason, a uniform downsampling is used in order to obtain the keypoints. Although this approach is not the best choice for rigid models, it works well with deformable models since it ensures a good distribution of keypoints along the surface of the object.

Once keypoints are selected SHOT descriptors are used to define each keypoint. SHOT descriptor shows a good balance between recognition accuracy and time complexity [**?**]. The SHOT descriptor encodes information on the topology of the surface in an area that stores information about the neighbourhood of a point. The area is divided into 32 bins, with 8 divisions along the azimuth, 22 along the elevation and 2 along the radius.

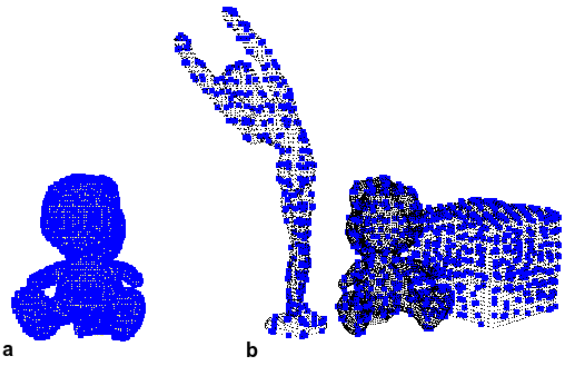


Figure 2: **a:** Reference model with all points as keypoints **b:** Scene that contains the reference model sampled with uniform downsampling to choose the keypoints. The keypoints are colored in blue.

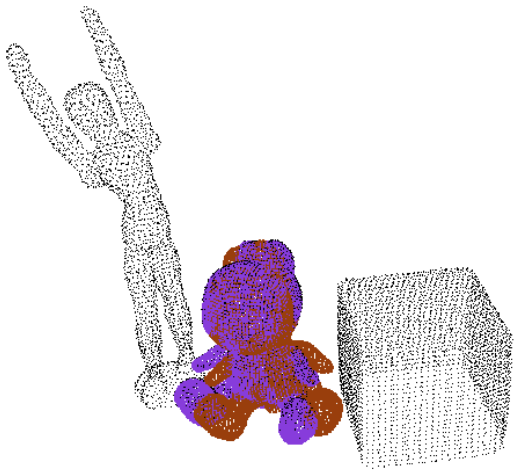


Figure 3: Two possible transformations corresponding to two instances of Fig.2(a). The purple transformation fit better than the brown transformation.

3.2 Rigid Detection Pipeline

The rigid detection pipeline starts with extraction of the keypoints and descriptors of the scene. This process is done using the same method as in the model initialization, i.e. using a uniform downsampling. But, in this case the frequency of the sampling is lower because of performance reasons (Figure 2).

Once the 3D keypoints and descriptors are computed, the descriptors are used in order to match the keypoints of the current scene and the keypoints of the reference model. All the correspondences obtained are grouped into subsets or instances. These instances are built enforcing geometric constraints between pairs of correspondences [**?**]. If there are not enough matches to allow a correspondence grouping, the scene is downsampled again iteratively increasing the sampling frequency.

After obtaining the set of instances, a rigid transformation is obtained from the instance with the higher number of correspondences (see Figure 3). The rigid transformation is computed as in [**?**]. It can be computed with a minimum of three points to obtain the position and orientation with 6 DOF.

This result is used to partially register the scene with the reference. However, points not belonging to the selected instance may still not be aligned if the scene has deformations. In order to detect this situation an inlier test is performed using the distance between the points of the partially registered scene and their correspondences in the reference model. For points classified as outliers the registration process is executed again iteratively. This approximation is very effective to represent those deformations which can be expressed as chain of rigid transformations. The iterative process stops when a maximum number of iterations is reached, or a fixed percentage of correct matches is obtained. These thresholds are configured depending on the particular problem domain, the number of deformations and the result of the deformation.

3.3 Non-Rigid Detection Pipeline

When the stop criteria is reached, the non-rigid detection phase begins with the non-matched points from the rigid detection phase as input. In this phase, the transformation of the best set of correspondence grouping is used to register deformed points near from corresponding points in the scene. So, this transformation is a first approximation to the place where finally the deformed points could be localized in the scene.

With the first approximation performed, a radius search is executed for each non-matched point. The search is based on a threshold used as max distance between each reference model point and its corresponding scene point. The point of the scene closest to each searched point of the reference model, is taken as correspondence of the point. In addition, only non-matched points of the reference model and scene are used for the phase of non-rigid detection. The rest of the points are not taking into account for this phase. It improves the point search time and reduces the possible false positive in the matching process.

4 RESULTS

In this section we present a set of three experiments that show the results obtained using the proposed method. The experiments are divided into two groups: synthetic experiments that measure the accuracy and performance of the proposed method under controlled conditions using cad models, and not synthetic experiments that are focused to evaluate the method using models and scenes obtained with depth sensors. The solution used to obtain the models and scenes to the last group

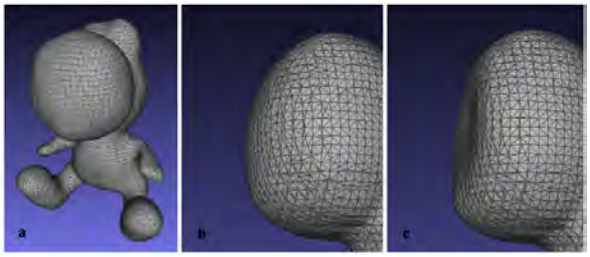


Figure 4: **a:** Reference model without deformations **b:** Detail of the candidate region to be deformed corresponding to the model **c:** Detail of the deformed region corresponding to the model

is Structure Sensor for mobile devices [**?**] with a simple 3D scanner. For all the experiments, only the point cloud corresponding to the vertices of the models and scenes are used in the method.

The experiments have been performed in a computer with Intel Core i5 3.2GHz, 8 GB of RAM DDR3 665MHz and Windows 8 64 bits operative system.

4.1 Synthetic Experiments

In the first experiment, the performance is evaluated using the model in Figure 4 (a). The model has 11798 points. The same model with a translation in one axis and two rotation in different axes is defined as model to be detected, so that the two models are misaligned. The aim is measuring the used time in the different detection tasks. The five main task involved in the process of detection are normal computation, sampling, descriptors computation, correspondences computation and correspondence grouping. Figure 5 shows the times for the different tasks against the sampling factor. The sampling factor is steadily reduced by 20% in each test, thereby increasing the number of keypoints used for the detection process.

The main execution time corresponds to the correspondences computation task. The normals computation task and the sampling task are constant with very low cost in terms of time. Moreover the descriptors computation, correspondences computation and correspondence grouping tasks increase proportionally to the number of points and therefore inversely to sampling factor.

For the second experiment, the model with the deformation in Figure 4 is inserted in a cluttered scene (see Figure 6). The scene is translated in one axis and is rotated twice in different axes to produce a misalignment with the model. Different level of noise is applied to each dimension of the 19836 points (see Fig6(a,b)). The noise has a uniform distribution between -1 and 1 that is multiplied by a maximum displacement for each intensity level of noise.

Figure 5: Time of five main task involved in the process of detection

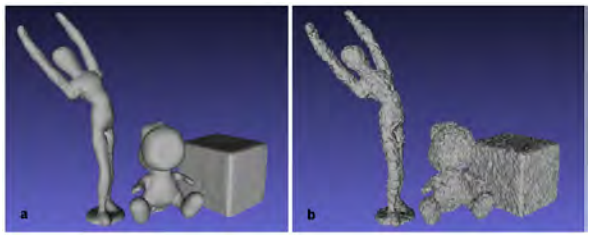


Figure 6: **a:** Original Scene (without noise) **b:** Scene with random noise in the vertices

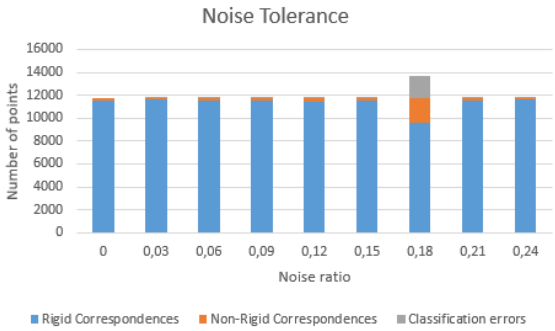


Figure 7: Characterization of the scene (Fig.6(a)) points in rigid or non-rigid with different percentage of word unit (0.0060702) used as max distance in each level of noise

Figure 7 shows the characterization of the different points in rigid or non-rigid region. The 0 column is the reference case with 11579 points corresponding to the rigid region and 189 point corresponding to the non-rigid region. The classification errors measure the number of points incorrectly classified respect to the reference case. The max displacement of each level of noise introduced to the scene is a percentage of the unit world (average distance of all points to its closest point) fixed in 0.0060702.

In most cases, the characterization of the point in rigid or non-rigid is correct and hence the matches are correct. Only the case with the 18% of the max displacement of the noise presents high classification errors, however the obtained matches are correct. Thereby the characterization of points which are wrong classified is incorrect but the matches obtained are correct.

The experiments with random noise demonstrate how robust the method is. While the models preserve the surface, it is possible to determine the deformation between the reference model and the scene with a reasonable error due to the noise.

4.2 Not Synthetic Experiments

The next set of experiments are performed to identify deformations using Structure Sensor to capture the reference model and the scene. The reconstructions obtained present noise but preserve the topological information of the object upon which the detection will be run. The following set of experiment is performed under the cited assumption.

As a general rule for the figures in the section, the green model represents the reference in the experiments. The blue lines, show a representative subset of the correspondences detected. A representative number of correspondences, and not all, are drawn for better visualization of the correspondences in the experiments.

Figure 8(a, b) shows the acquired pillow model used as reference. In this case a soft pillow is used. On the other hand, Figure 8(c,d) shows the second acquired pillow model with a deformed region produced by a force applied in the centre of the pillow.

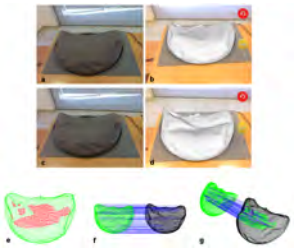


Figure 8: **a,b:** Reference model without deformations **c,d:** Model with deformed regions **e:** Reference model with the deformed region in red color and undeformed region in green **f:** Rigid correspondences **f:** Non-rigid correspondences

The method detects the deformed and undeformed regions (see Fig.8(e)), using as input the pillow reference model from Figure 8(b) and the second model from Figure 8(d). For each point of undeformed region (see Fig.8(f)) and for each point of deformed region (see Fig.8(g)), the correspondences between reference and the model are calculated. The green points of the pillow in Figure 8(g) represent the points of the scene detected as corresponding points of the deformed region. The execution time is about 4.55 second using 2067 keypoints of 10336 points of the reference model and 2654 keypoints of 10380 points of the second model.

Using the same reference model as in the previous experiment, the experiment is performed in cluttered scene (see Fig.9(a,b,c,d)). Figure 9(a,b) shows the state of the scene before the pillow deformation, and Figure 9(c,d) shows the state of the scene after the pillow deformation performed in the centre of it. The reference model has been segmented from the reconstruction obtained in Figure 9(b).

The distinction between deformed and undeformed is displayed in Figure 9(e). The point matching for undeformed regions are shown in Figure 9(f) and the deformed regions in Figure 9(g). The execution takes about 5.48 seconds using 2425 keypoints of 6015 points of the reference model and 2926 keypoints of 14478 points of the scene.

4.3 Discussion

The sampling for the model and the scene directly influences in the time execution and in conjunction with the threshold used in the rigid detection pipeline are main sensible parameters. Both determine the goodness of the result and depend on the resolution and on the characteristic topology of the reference model. Bad parametrization of the values produces bad characterization of the some points like deformed points, but nevertheless the match between the reference model and the scene is good.

The proposed method does not work with full deformation or greatly exaggerated deformations and fails if it is

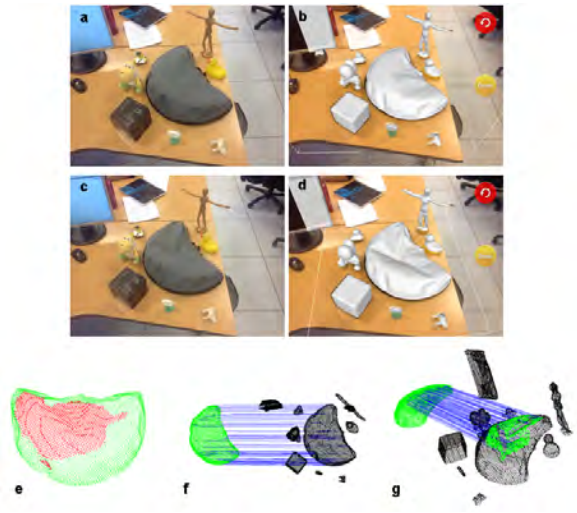


Figure 9: **a,b:** Reference model without deformations in a cluttered scene **c,d:** Model with deformed regions in a cluttered scene **e:** Reference model with the deformed region in red color and undeformed region in green **f:** Rigid correspondences **f:** Non-rigid correspondences

folded upon itself. It is necessary a region undeformed, large enough compared to the reference for searching the possible deformed regions. Also fails with models that are not topologically characterizable or without enough surface characterizable. An pragmatic example of this case is a sphere. It is impossible to know which points have been exactly deformed because any section of the surface is identical to any other section of the sphere.

When the objects present joints also can be approached as a chain of transformations (see Fig.10). In Figure 10, the reference model has two deformation produced by two rotations in two different parts of the humanoid, one in the waist and other one in the left elbow of the humanoid (see Fig.10(a,b)). In Figure 10(c), the two deformations respect to the reference model are detected and their corresponding points are matched in Figure 10(d).

In general the execution time is less than 1 second when the sampling factor is not too small and it increases the number of keypoints in the detection process. Thereby it might be possible the real-time execution, selecting the suitable values for the sampling factor and rigid detection threshold.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a 3D registration framework for deformable object tracking that can easily detect rigid and deformed regions. We propose the use of correspondence grouping that allows to

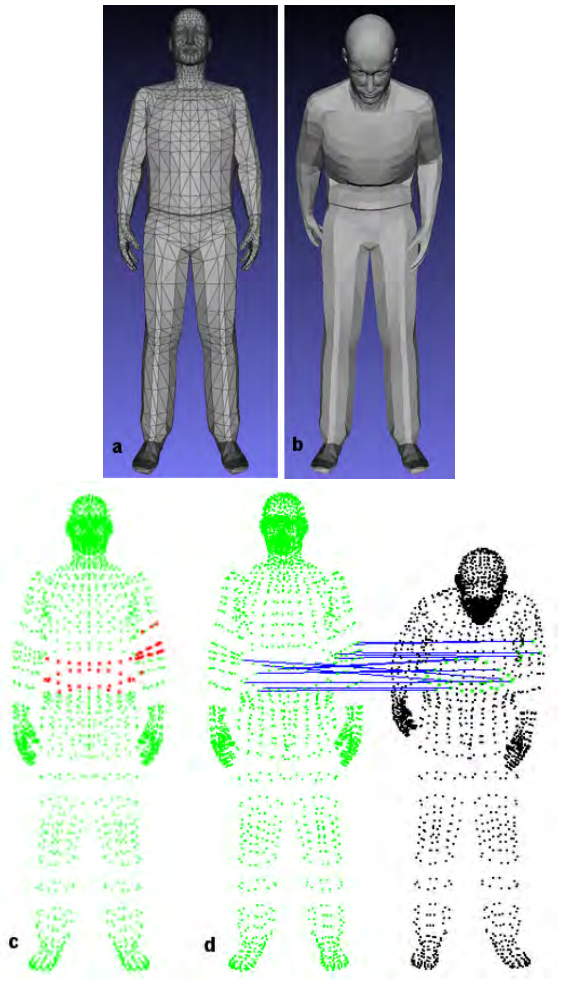


Figure 10: **a:** Reference model without deformations **b:** Model with deformed regions **c:** Reference model with the deformed region in red color and undeformed region in green **d:** Non-rigid correspondences

obtain a chain of rigid transformations for undeformed regions. This solution allows detecting deformed regions in a straightforward manner using a simple radius search. By this way, each point of the deformed region is matched with the closest point in the reference scene.

Unlike the solutions found in the state of the art, our approach relies in a simple 3D point correspondence strategy that allows converging fast and at the same time avoiding local minima.

Experiments have shown that the method behaves properly in cluttered scenes and it is particularly suitable for point clouds captured using commodity depth cameras. Moreover, the set of experiments performed in an uncontrolled environment proves the validity of the method. The method make it possible to isolate the undeformed regions and search for the deformed regions. Additionally, the set of experiments concerning to the chain of deformations shows that it can be suitable to obtain a chain of rigid transformations wherever needed.

As future work it is planned to extend the method integrating a frame-to-frame tracking strategy. This would allow to get a more stable and faster convergence, avoiding to compute in each frame all the point correspondence grouping. Moreover it would also make it easier to filter the input cloud detecting more outliers.

Finally, the work exposed is the first step in the development of a system for modelling elastic objects using physical mass-spring simulation techniques. Once completed, the registration could be further improved introducing mechanical constraints to the proposed instance grouping algorithm.

6 REFERENCES

[Ale12] Luis A Alexandre. 3d descriptors for object and category recognition: a comparative evaluation. In Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, volume 1. Citeseer, 2012.

[AMT+12] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Point cloud library. IEEE Robotics and Automation Magazine, 1070(9932/12), 2012.

[AVB+11] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bodan Rusu, and Gary Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pages 585-592. IEEE, 2011.

[CB07] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. Pattern Recognition Letters, 28(10):1252-1262, 2007.

[FHK+04] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas BÃ$\frac{1}{4}$low, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In Computer Vision-ECCV 2004, pages 224-237. Springer, 2004.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM symposium on User interface software and technology, pages 559-568. ACM, 2011.

[LAGP09] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. In ACM Transactions on Graphics (TOG), volume 28, page 175. ACM, 2009.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91-110, 2004.

[LSP08] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In Computer graphics forum, volume 27, pages 1421-1430. Wiley Online Library, 2008.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, pages 127-136. IEEE, 2011.

[Occ15] Structure Occipital. Sensor for movile devices: http://structure.io, February 2015.

[RBB09] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 3212-3217. IEEE, 2009.

[RBTH10] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 2155-2162. IEEE, 2010.

[RMBB08] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Persistent point feature histograms for 3d point clouds. In Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany, pages 119-128, 2008.

[Rus10] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. KI-KÃ¼nstliche Intelligenz, 24(4):345-348, 2010.

[Rus15] Radu Bogdan Rusu. Point cloud library (pcl): http://pointclouds.org, February 2015.

[SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In Symposium on Geometry processing, volume 4, 2007.

[SF10] Mathieu Salzmann and Pascal Fua. Deformable surface 3d reconstruction from monocular images. Synthesis Lectures on Computer Vision, 2(1):1-113, 2010.

[SRKB10] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), volume 44, 2010.

[TSDS10] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In Computer Vision-ECCV 2010, pages 356-369. Springer, 2010.

[WV11] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3d object classification. In Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, pages 2987-2992. IEEE, 2011.

[Zho09] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 689-696. IEEE, 2009.

[ZNI+14] Michael Zollhofer, Matthias Niebner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. ACM Transactions on Graphics, TOG, 4, 2014.

Abstract Surface Modeling for concurrent Form Finding and Class A Surfacing in Computer-Aided Design

Sebastian Misztal
University of Applied Sciences and Arts
Ricklinger Stadtweg 120
30459, Hanover, Germany
sebastian.misztal@hs-hannover.de

Ingo Ginkel
University of Applied Sciences and Arts
Ricklinger Stadtweg 120
30459, Hanover, Germany
ingo.ginkel@hs-hannover.de

ABSTRACT

As the missing link between designers and engineers, we introduce a new abstract modeling approach for computer-aided design systems. In contrast to existing solutions, our strategy is less geometry driven and less based on low-level aspects like control points or mesh elements. Instead we operate on the idea of a hierarchical modular concept with abstract components like categories, parts and the features relations. The whole surface structure of the model is composed of abstract areas represented by meshes. This allows designers without engineering background to concentrate intuitively on the form finding process as they easily model abstract components and automatically generate high quality CAD-freeform equivalents suitable for computer-aided manufacturing. During the phase of construction, we focus on the designers intent and guide him through this process to enrich the model with semantic information. The goal is to describe the models structure, such that the automatically generated freeform surfaces not only meet correct geometry but also mirror the internal configuration of the model. Compared to common practice where design changes and updates lead to the reconstruction of the whole model, our system accomplishes these kinds of alterations automatically, based on the hierarchical model configuration, derived from the designers intent. So our approach of an abstract modeler is much faster, closes the gap between creative design changes and technical model construction and captures this in one contemporary system and workflow.

Keywords

Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM), Geometric Modeling, Interaction Techniques, 3D Modeling, Class A Surfacing, Shape Design, Object Representations

1 INTRODUCTION

Designing and manufacturing a product incorporates various tasks to be performed by professionals with partly very divergent backgrounds. These can be roughly classified into designers with creative minds on the one hand and engineers with technical expertise on the other hand.

Designers, who are responsible for form finding, often refuse technologies like splines and NURBS. So they usually rely on 2D-sketches, real clay models or virtual mesh models of the object they are creating.

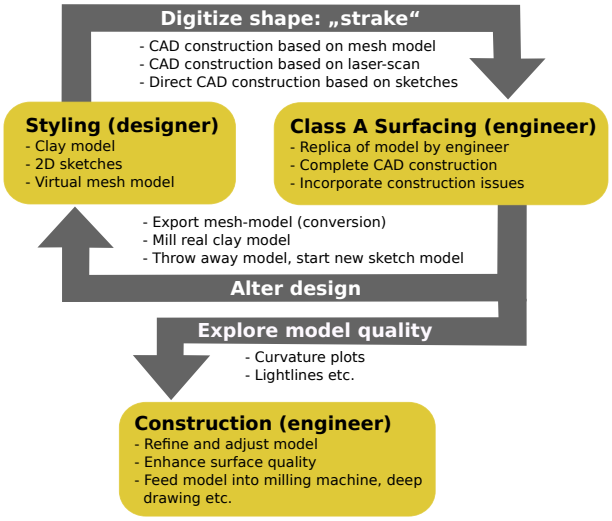


Figure 1: *Common design / CAD workflow*

Engineers have to transfer the model into a manufacturable model in a further step. Based on a real clay model or sketches, the whole technical CAD-model has to be created. In the case of a virtual mesh model one has to state that meshes can not provide a suitable mathematical surface quality, thus a CAD-replica of the shape is also necessary.

So for all of the usual design approaches, a hand-crafted recreation of a whole model into a freeform surface representation is everyday business and of course very time-consuming. Even more serious is the problem when a designer wants to alter the shape of a model afterwards and the engineer has to rebuild the whole freeform model again. Reverse eingineering in different process steps became mandatory [Sok05]. The media disruption between a model created by a designer and the one formed by an engineer and its impact to the overall workflow is depicted in figure 1.

The task to overcome the media disruption is either to teach an engineer a creative mind or to provide designers a suitable modeling tool for designing and producing a manufacturable model simultaneously. The goal of our approach is the latter one, providing the designer an intuitively usable tool which guides him through the modeling process and tries to capture the design intent. Since neither meshes nor spline representations are suitable for a designer to simultaneously shape the desired object and create a manufacturable model, a combined or hybrid approach is necessary.

The idea is to build a system which allows modeling on a less detailed but hierarchical abstraction level. The model will both cover a mesh model and a spline representation but neither of them will be modified directly in the sense of low level polygonal transformations, control point movement, degree selection or parametrization changes. This kind of construction is too much a low level approach and distracts the designer from his main task, namely to find the overall shape of the object.

Instead we provide the user with an abstract surface model consisting of abstract surface areas, hierarchical dependencies and enrich this model with semantic information which will be queried from the user or can implicitly be derived from the context (i.e. the used construction tool, moment and region where and when it is applied). Topological operations will not be performed on meshes or spline surfaces, but on an abstract level containing adjacency between abstract areas freed from low level aspects like mesh consistency or curvature continuous transitions.

Based on this abstract model, a mesh surface for preview purposes will automatically be generated delivering a fast feedback. In contrast to a usual mesh model, additional sematic information, design intents etc. are still stored in the abstract model. After some iterations

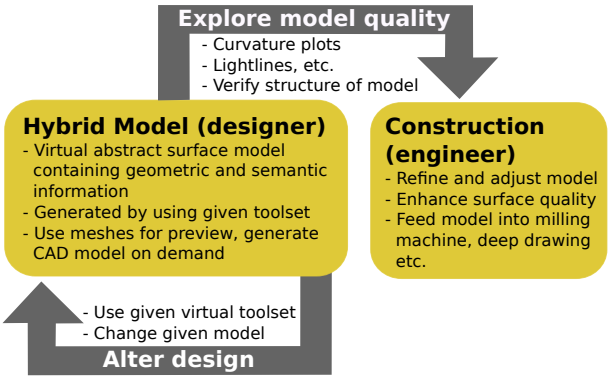


Figure 2: *Intended design / CAD workflow*

of design changes and form finding, this information can be used to automatically create a class A surface model which can be used as input for the construction phase.

Eliminating the media disruption between the design- and the manufacturing-model (i.e. designing and constructing in a joint hybrid model) the overall workflow can be simplified substantially, as shown in figure 2.

2 MODELING CONCEPTS AND TECHNOLOGIES

Modeling objects using virtual construction tools basically allows three conceptually different approaches. The first one are mesh-modelers, providing the generation of simple geometric shapes. Another solution are CAD-construction tools using freeform surfaces for class A surfacing. The third concept are solid modelers, focussing on solid primitives, building constructive elements. In any of these configurations a designer and an engineer still produce the before mentioned media disruption.

Most CAD/CAM tools focussing on the media disruption between design and construction phase try to simplify and automate the conversion between the generally used representations. This also holds for 3D sketching approaches like in [Zel06] or [Die04]. These systems are trying to guide the designer through a model finding process, but still require a conversion of the design- (i.e. the sketches) into a construction-model. Using a converter is of course much faster than hand-crafting the model again for construction, but converters often demand adaptions after every conversion. Additionally with every conversion, valuable information get lost. Converting from a design-representation to a construction model, design intents and specific shapes can be sacrificed for technical restrictions. Converting a construction model into a design model often causes a loss of structural information.

To recover these informations various startegies are pursued, e.g. shown in [Han00]. Reverse engineering ap-

proaches traverse the history-tree or -graph and thus all recorded modeling operations and interpret their constellation to extract valuable meta information from it. But this requires an expressive tree- or graph-structure, holding the desired intent. Hint-based solutions analyse geometrical characteristics which also only qualify for capturing geometrical features. Knowing the pure features without any context is insufficient knowledge to create an overall view of the designers purposes. Moreover, relations are often not cosidered.

Concepts like [Li10] divide the whole modeling arrangement into smaller sub-parts to find symmetries which makes it easier to detect features and the design intent. Conceivable is also to split the history tree into little pieces too. The limitations of history-trees is the fact that they map a chronological sequence. In contrast, our idea is independent from the order of the operations by just using the structural and hierarchical information.

Establishing a modeling methodology like [Bod14] with advanced rules, predetermining the sequence of types of operations during the modeling process, lead to a clearer history-tree and a better mapping of relations between elements and thus to better conditions for mentioned intent recognition approaches. But this methodology still requires a strong contribution of the designer and is less technology driven like the strategy of [Ald12] and [Dor13]. These attempts demand the designer to make annotations, describing their design intent which is helpful in further modeling steps especially for other designers, working with an unfamiliar model. Unfortunately these annotations are interpreted by human beeings and not by a machine.

Machine-interpretable technologies to capture the design intent are given in [Kim06] by reasoning which is ontology-based. Spatial relationships can be stored in a relational model. But the focus here lies on assembly design which is difficult to adapt to classical product design issues.

During the evolution of CAD systems, different generations and paradigms emerged, dealing with the capturing of design intents in the broadest sense, e.g. presented in [Tor10]. Parametric technologies and feature-as well as history-based systems in solid modelers are the fundamental idea, capable to describe the model precisely enough to map geometrical relations and dependencies. One of the major drawbacks is the steep learning curve. Direct or also explicit modeling tools have the advantage of beeing very simple, easy to learn and fast. But the absence of a construction history and missing relations require reverse engineering to capture features and design intents.

So today vendors tend to incorporate both approaches to hybrid solutions in its literal sense. That means that these concepts are not completely combined. Instead they often operate synchronously where the designer can switch between them like in *Siemens NX and SolidEdge*. *PTC Creo Parametric* also allows the adding of constraints to the model during direct modeling. But our goal is to create high quality freeform surfaces straight from the explicit modeling object. Other professional software like *Autodesk Fusion 360* provide tools like *snapping* which allows the user to create the freeform surfaces directly on the mesh but still being a time-consuming low-level attempt. Declared converters like *Kontenpunkt PointMaster* produce correct freeform surfaces but neither capture the designers intent nor provide an internal model structure that is suitable for production without further handwork. Solid modelers like *SolidWorks* provide very nice internal structures of the model, containing defined relations between parts. They also provide precise shapes of primitives based on implicit representations which are previewed by meshes. Their weak spot is the restriction to fairly simple surfaces based on the solids. So creating a real freeform surface with a solid modeling tool is very challenging.

To construct our system, we borrow ideas from each of these approaches. We use meshes for preview, parametric surfaces for construction models and an internal structure of abstract regions that are aligned and interact very similar to a solid modeling concept. This tool will allow arbitrary surface shapes, explicitly modeled internal dependencies and hierarchies built by a designer with limited technical skill or interest. The goal of our approach is to guide and help the designer to intuitively find the form or shape of an object. Simultaneously we need to establish the preconditions that allow an automatic retrieval of construction data, namely a high quality CAD model covering both the shape and internal structure (part groups, hierarchy etc.).

3 ABSTRACT MODELER

Our abstract modeler is an extending modeling paradigm which is theoretically transferable to various modeling systems. We integrated our technical implementation into the CAD-system *Rhinoceros 5* as a plug-in. We used the existing surface representations and tools as low-level objects and operations respectively and created our own high-level objects and operations upon them. In this section we introduce the basic concepts of our paradigm and describe the structure of our solution by an exemplarily workflow.

3.1 Modeling Elements

Meshes are consistent structures where the features are characterized by an absence of mesh elements or by a specific geometry or constellation of vertices. That means that features in a common mesh are not just integrated into the whole construct, they are melded into

the mesh. Freeform surfaces expose their features in a similar manner. They are defined by the mathematical representation of the surfaces and finally a geometrical declaration. Features in both technologies basically do not differ from the rest of their object, as the border between a feature and the rest of the model is fluent. An automatized way to identify a feature would be very expensive as they have to be retrieved afterwards by using complex algorithms. And there is no guarantee that all features and the hard boundaries between a feature and the rest of the model will be found.

Nevertheless our system is based on the idea of a modular concept with separable elements with a clear distinction between them. The following properties state our requirements to these elements:

- Each feature or part of the model is an individual, distinct and nameable model-element.
- Each model-element can be addressed and selected.
- A model-element has a designated state, carrying meta information.
- The meta information of a model-element precisely describes its own geometry and its relation and connectivity to other model-elements.
- A model-element is a module, which can be exchanged and integrated into another model.
- The consistent surface of the model is formed through the connections of a set of model-elements.
- Operations can either be performed on an element individually or by a related set of elements using their meta information and geometry.

As mentioned before, common surface elements like meshes and freeform surfaces are not the fundamental objects in our attempt to describe a model. The elements we use are an abstract generalization not only storing geometrical information. The visual shape of a model is composed of the two model-elements *additive* and *area*.

Definition 1 (Area) *An* area *is a distinct, bounded area, lying directly on the geometrical surface of a model. Together connected, areas form the outer hull of a model.*

Areas are the basis element comparable to faces of meshes or freeform surfaces in typical CAD-models but with fundamental differences. They do not form the whole model. Instead they shape the coarse form of it and function as a carrier for other elements. In practical application, areas not only have the task to structure a model but also to organize other elements of the model. To describe the whole model, there is a further element called additive.

Definition 2 (Additive) *An* additive *is a combination of arbitrary geometrical elements. As a closed unit, additives are modules with the ability to be attached to areas. An additive itself can again be composed of a modeling structure with areas and additives.*

Additives represent a self-contained feature. In practical application, whenever a designer wants to create a part of the model which can be named and modeled on its own, detached from the rest, he would create a new additive. As areas only describe the coarse parts of the model, additives are used when fine sections have to be created.

Definition 3 (Layer) *Each area can be layered. A* layer *is a specific area with its attached additives in a system with under- and overlaying other areas with their additives. A layer has a level, synchronous to its time of creation.*

Basically a layer is not more than a specific decorated area instance which can be exchanged by other layers. In practical application, several layers are used when the designer wants to experiment with different shapes and additive constellations. As each layer has a level, according its chronological creation, the designer can travel the progress time by switching the level of the layer. The unique characteristic here is the fact that individual parts of the model can be layered and traversed. Thus it is possible to assemble and modify a model with parts of several progress steps.

Definition 4 (Base Object) *The* base object *is a compound of not overlapping areas, on not necessary equal leveled layers, forming a gapless (and solid) surface structure. Each not overlapping (and solid) constellation of areas forms a valid base object, representing one instance of the models surface.*

Definition 5 (Model) *A* model *is the entirety (set) of all model-elements (additives and areas), layers, relations and operations, allocated to this model.*

Figure 3: *Engine hood of a vehicle with attachments.*

Figure 3 shows a part of a vehicle with different model-elements. The surface of the model is constructed of areas (colored in yellow), all together forming the base object. Different shades of yellow indicate different layers. The spare tire and the headlights are designed as additives (colored in red and complete models themselves) attached to the engine-hood-area. The connectivity of the spare tire to the engine hood is highlighted in green. The position of the tire can be adjusted by using control points (also in green). Moving these control points alters the areas geometry as the hole for the tire is moved too. Control points at the border of the middle area adjust the size of this area which in turn has influence on the tire depending on the relation between them. Varieties of this impact are illustrated in figure 4.

3.2 Meta Information

As all model-elements are defined as separable objects, relations are the connecting element between them, creating the consistency of a model.

Definition 6 (Relation) *A* relation *is a meta information, describing the state between two model-elements with the following parameters:*

- *The connectivity, including the type (loose, fixed, fluent, static etc.) and adjacency.*
- *The dimension, including the sizing information.*
- *Distances, including absolute and relative lengths.*
- *Repetition, including the mirroring of elements using determined distances.*
- *The rank, including priorities of elements and relations.*

In practical application, relations come into play when the model or single parts of it are modified. These relations describe the behavior of linked elements and the modified object itself.

Definition 7 (Operation) *An* operation *is an action directly applied on one or more model-elements, areas and/or additives. Operations are all kinds of transformations on an element, not changing its relations. Relations of the considered and related elements are triggered through an operation. Operations aggregate the type of transformation, its location and the applied tool with parameters.*

Unlike relations, an operation is not a determined and fixed state. Operations are sequential and retraceable actions where the order matters. In practical application, operations are used to shape an element.

Definition 8 (Mode) *A* mode *is a state, chosen by the user, defining the impact of an operation on the model by redefining its relations.*

The idea is to attain different effects with the same operation just by switching the mode. Theoretically our systems allows an unlimited number of modes, some pre-defined by the system, further modes can be created by the user himself. We introduce two fundamental modes, also used in our example. The *geometry-mode* is a state where geometrical operations on a selected element have pure and isolated geometrical impact only on this particular element. The designer is performing plain low-level geometrical modifications on separated parts of the model without further impact on other parts. This is the kind of behavior which a designer would expect as he knows it from familiar modeling tools. Here the geometry-mode is mostly used for shaping the outer form of the model or isolated features. The *semantic-mode* is a state where geometrical operations on an element not only modifies its geometry but also all parts of the model which are semantically connected to it. As discussed before, adjacent elements of the model can be mutually related. These relations are used to determine the effect of an alteration.

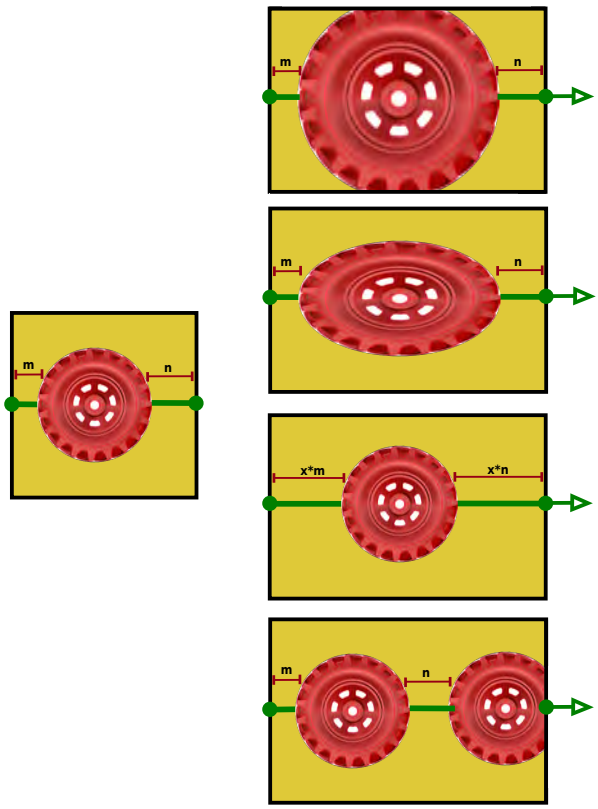


Figure 4: *Tire as an additive, attached to an area.* ***Left:*** *Initial state.* ***Right:*** *Various impacts on the tire after resizing the area depending on their relations.*

Figure 4 illustrates a tire on an area, similar to the example in figure 3. The tire is attached through relations between the border of the area and the border of the tire with defined distances *m* and *n*. When the area is enlarged (right column), the impact on the attached tire

depends on the predefined relations between them, particularly the parameters of the connections. The resizing of the area results in a resizing of the tire by satisfying the absolute distances *m* and *n* and its shape (*first image*), in a deformation of the tire by satisfying the absolute distances *m* and *n* (*second image*), in a translation of the tire by satisfying the relative distances *m* and *n* and its shape (*third image*) or in a mirroring of the tire by satisfying the absolute distances *m* and *n* and its shape (*fourth image*).

3.3 Mesh Display

Our surface representation is an abstract construct which needs to be displayed in some way. So to render our abstract surfaces we use polygonal meshes created by tessellation techniques from Delaunay [DeB08] and extended methods from Chew [Che87] and Shewchuk [She05] called Constrained Delaunay. This kind of tessellation creates triangles whose interior angles all tend to have the same size as they vary just a minimum user defined value from $\frac{\pi}{3}$. The result are homogenous looking meshes which is not only an advantage during rendering. The structure of the faces of these meshes also leave a more valuable impression for the designer.

Besides the pure display and tessellation, our system also uses subdivision techniques from Catmull-Clark [Cat98] and Loop [Loo87] and furthermore extended methods from Ginkel [Gin06]. Curved surfaces are created by subdividing the mesh representation of the areas to be shaped. Therefor the abstract area is decorated with an subdivision operation. These operations are editable and exchangeable. Other subdivision methods can be performed on the area by changing parameters.

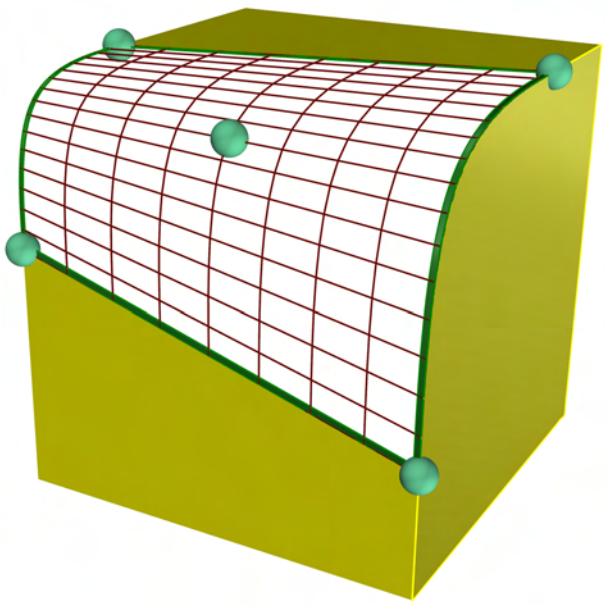


Figure 5: *Blended transition area.*

Figure 5 exemplarily shows the creation of a transition between areas using Catmull-Clark subdivision techniques. The designer can expand this blended area through four control points at the corners and alter its roundness through another control point in the middel. The border is highlighted in green color. Ordinary meshes or freeform surfaces would provide a much larger number of vertices or control points respectively. We use just five. The techniqual details are hidden from the designer to reduce complexity.

3.4 Micro Modeling Workflow

To create a model with elements satisfying the before mentioned characteristics, we imply this coarse modeling process in three steps:

1. First of all the designer has to be aware of what exactly he wants to do in each modeling step. Therefore our system guides the designer by showing possible workflow sequences and suitable previews of all operations.

2. After the designer disclosed his decision, each action must then be declared by him. Not by performing low-level operations but instead by triggering abstract high-level operations which are capable to classify the designers action and to record the necessary meta information. These high-level operations can achieve the same geometrical result as familiar low-level operations. But they have a different structure and procedure where the designer instructs the system to do the operation and does not do it by his own. These operations are meaningful and summarize a set of single actions.

3. Because we do not want the designer to state all necessary meta information by himself, which would end in a long querying sequence, the system responses high-level actions with appropriate default solutions. Afterwards the designer can adjust these solutions by altering the meta information interactively.

During the whole process a lot of meta information is collected but not all of it is entirely provided by the user. A lot of it arises implicitly by choosing a tool, by applying an operation or by appropriate default settings. Another speed-up comes from the interchangeability and automation of our workflow, when the designer decorates new or empty parts of the model with already applied operations and previously stored categories.

In our interpretation of a modern CAD-system the role of the designer changes. Unlike common systems which are often geometry driven, our concept does not follow the *What You See Is What You Get* approach entirely. By restricting low-level attempts on the

surfaces geometry and emphasizing the use of abstract operations, the designer becomes more an instructor. We picture the working procedure of him more like drawing a construction plan and less in forming each single shape and feature manually. Besides the creative form finding process, we mainly want the designer to enrich the model with semantic information which distinguishes us from other modeling systems. In the following we will show how this can work in an exemplary workflow.

3.5 Exemplary Workflow

In this section we describe the concept behind our idea through a typical workflow by means of the example of a washing machine. Going through all modeling steps, we introduce all kinds of elements, operations and our fundamental layer system. The model was kept deliberately simple and could also be created by a constructive solid geometry (CSG) system. But our methods are conceived with the aim to construct freeform surfaces equivalent to the model.

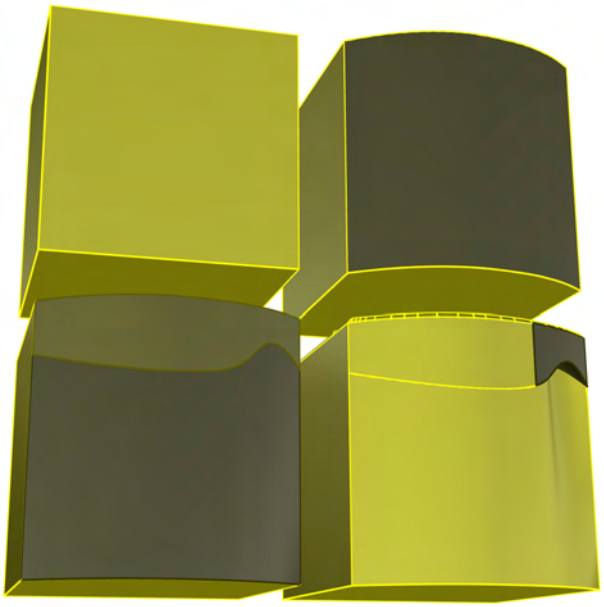


Figure 6: ***Top-left:*** *Base object;* ***Top-right:*** *Bended front area;* ***Bottom-left:*** *Front-area divided into panel-area (overhead) and door-area (below);* ***Bottom-right:*** *Panel-area divided into panel- and detergent-area. Deformation on door-area for the detergent dispenser and deformation on the top-area where the transition between the top-area and the rest was blended.*

Modeling Structure

The structure of our model is designed as a modular system with the main focus on interchangeability. On top of that system and at the very beginning of the workflow, there is the *base object*, the initial instance to work with, representing the coarse shape of the desired model. In our example the base object is a cube (figure 6). Each of its sides represent a specific user defined part on the surface, the *areas*. An area can be shaped like shown in our example where the designer bended the front area of the washing machine. From then on a bending operation is assigned to the front-area. But the main aim of areas is to organize the elements upon it. Therefor the designer can divide an area – like the bended front-area – into two separated areas. Each of them can then be modified individually like the door-area which was dented for the detergent dispenser.

Attaching Additives

Figure 7: *Attached additives colored in red.*

Furthermore an area can host supplementary features, the *additives*. Such an additive can be an entire model itself with a base object and areas or a plain geometrical object, depending on its complexity. In our example (figure 7) the panel-area is decorated with additives representing a display, some buttons, a rotary control and a coating for the detergent dispenser. Other additives are attached to the door-area for the door and the doorhandle.

Relations

The areas are connected through relations by default. All attached additives have at least one relation to its underlying area or other additives, describing its connectivity and behavior. In figure 8 the relation between the rotary control and the panel-area is displayed. This additive is connected to its area with a static connection, which means that in case of a deformation of the area, the distance between the additive and the areas border stays constant. The buttons on the panel-area are connected to the areas border and with each other. Control points adjust distances and the location of attachment.

Layer-System

Apart from their geometrical form, and their function as a host for additives, areas are predestinated to organize these additives by grouping them. But areas can also be layered. Like shown in figure 6 the door- and the panel-area were layered upon the front-area which is still available and not deleted. Moreover these layered areas are logically connected. The designer can flick through these layers and modify a certain one which has direct influence on the other layered areas automatically. This gives him the ability of saving a snapshot

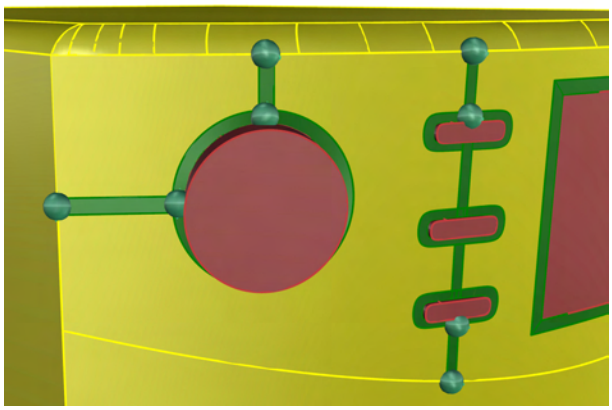


Figure 8: *Relations between additives and their area with control points.*

of a concrete part of the model for testing purposes or other design playthings. So different layered areas can be shaped differently and decorated with different additives, fully modular. As these areas can easily be exchanged, the designer can try out various constellations of the model in little amount of time.

This is another crucial distinction to other modeling systems. Going back in time in our solution means changing a layer and not changing the memory state. Thus a modification in the past has automatic influence in the future because of the relations between the layers. Layers give the designer the opportunity to just time travel selected parts (e.g. the front of the washing machine) and leave the rest of the model untouched. So various states of operations from various points in time can easily be combined.

Modeling Modes

The impact of an operation also depends on the chosen modeling-mode. In figure 9 the designer decided to modify the shape of the panel- and the door-area. On the left picture we see the old model. The right picture shows that he wanted the border between these two areas less curvy. So he modified the geometry of the border curve which is the coupler between both areas. Because this modification is conducted in semantic-mode, not only the geometry of this particular element was changed, also all relations to this curve are involved, including the adjacent areas and their attached additives. This means that obviously both areas were altered. But also the door was relocated, still fitting the same distance to the border and still retaining its size. Same goes for the buttons. The display was resized as the users intent defined it has to satisfy a fixed distance to the borders of the panel-area.

Figure 9: *Semantic-mode: Altering the shape between two areas.* ***Left:*** *Old shape with a curvy coupler between the areas.* ***Right:*** *New shape with a less curvy coupler.*

Figure 10 shows a doorhandle which was modified in geometry-mode (left: old state, right: new state). This operation has no impact an other elements and the alteration on this additive was only applied on its geometry. Here we can see both modes in contrast and how the same operation in different modes varies in their impact. In figure 9 the same operation in another mode also changed the whole arrangement of the additives. Whereas an operation in semantic-mode incorporates all gathered meta information, the geometry-mode only considers the pure geometry.

Figure 10: *Geometry-mode: Geometrical modification on the doorhandle.*

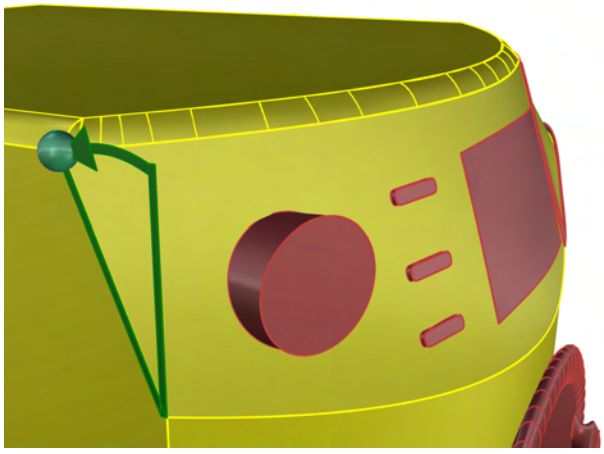


Figure 11: *Semantic-mode: Tilting the panel area.*

Another use case is shown in figure 11 where the panel-area was tilted and the attached additives on that area were automatically tilted too. The rotary control was automatically extruded, fitting its basis plane from before the modification.

4 SUMMARY AND FUTURE WORK

We have presented a modeling approach that enables a designer to construct a 3D model by building shapes and internal structures on an abstract level. Previews are accomplished by meshes and subdivision techniques while derivability of a CAD model based on the hierarchical structure of the abstract model is still guaranteed. The designer is equipped with construction tools and guided through the process of creating an object. For now we have focussed on the creation of the model starting from a base model. The presented tools were mainly meant to add features, both in geometric and in an abstract sense.

In the future we are going to extend the functionality by tools that modify the shape of a surface area. First to mention there is deformation. Neither control-point moving in a spline sense nor mesh deformation techniques in the sense of smoothing operators will be suitable to be used by a designer. Again we need a more abstract and non-technical view to the task. A surface deformation tool must be imaginable in a designer-context and could for example be interpreted as adding or removing clay from a certain surface region or to apply pressure to a rubber surface. Depending on which level our hierarchical model is attached to the surface area, we need to impose characteristics and restrictions to the allowed operations and of course to transfer the abstract modeling operator into a low level mesh and/or spline equivalent. Again geometric deformation and structural aspects will have to be executed simultaneously on the abstract model, the mesh and the spline surface-model.

5 REFERENCES

[Ald12] Alducin-Quintero, G., Rojo, A., Plata, F., Hernandez, A. and Contero, M. 3D Model Annotation as a Tool for Improving Design Intent Communication: A Case Study on its Impact in the Engineering Change Process. Proceedings ASME 45011, Volume 2: 32nd Computers and Information in Engineering Conference, 2012.

[Bod14] Bodein, Y., Rose, B. and Caillaud, E. Explicit reference modeling methodology in parametric CAD system. Computers in Industry 65 (1), 2014.

[Cat98] Catmull, E. and Clark, J. Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. Seminal Graphics, 1998.

[Che87] Chew, L.P. Constrained Delaunay Triangulations. Proceedings of the Third Annual Symposium on Computational Geometry, 1987.

[DeB08] de Berg, M., Cheong, O., van Kreveld, M. and Overmars, M. Computational Geometry - Algorithms and Applications, Third Edition. Springer, 2008.

[Die04] Diehl, H., Mueller, F. and Lindemann, U. From raw 3D-Sketches to exact CAD product models - Concept for an assistant-system. Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling, 2004.

[Dor13] Dorribo-Camba, J., Alducin-Quintero, G., Perona, P. and Contero, M. Enhancing Model Reuse Through 3D Annotations: A Theoretical Proposal for an Annotation-Centered Design Intent and Design Rationale Communication. ASME 2013 International Mechanical Engineering Congress and Exposition, 2013.

[Gin06] Ginkel, I. and Umlauf, G. Loop subdivision with curvature control. Eurographics Symposium on Geometry Processing, 2006.

[Han00] Han, J., Pratt, M. and Regli, W.C. Manufacturing Feature Recognition from Solid Models: A Status Report. IEEE Trans. Robotics and Automation 16 (6), 2000.

[Kim06] Kim, K.-Y., Manley, D.G. and Yang, H. Ontology-based assembly design and information sharing for collaborative product development. Computer Aided Design 38 (12), 2006.

[Li10] Li, M., Langbein, F.C. and Martin, R.R. Detecting Design Intent in Approximate CAD Models Using Symmetry. Computer-Aided Design 42 (3), 2010.

[Loo87] Loop, C.T. Smooth Subdivision Surfaces Based on Triangles. Master Thesis, University of Utah, 1987.

[Mun03] Mun, D., Han, S., Kim, J. and Oh, Y. A set of standard modeling commands for the history-based parametric approach. Computer-Aided Design 35 (13), 2003.

[She05] Shewchuk, J.R. General-Dimensional Constrained Delaunay and Constrained Regular Triangulations, I: Combinatorial Properties. Discrete and Computational Geometry 39, 2005.

[Sok05] Sokovic, M. and Kopac, J. RE (reverse engineering) as necessary phase by rapid product development. Journal of Material Processing Technology, 2005.

[Tor10] Tornincasa, S. and Di Monaco, F. The Future and the Evolution of CAD. 14th International Research/Expert Conference, 2010.

[Zel06] Zeleznik, R.C., Herndon, K.P. and Hughes, J.F. SKETCH: An Interface for Sketching 3D Scenes. ACM SIGGRAPH Courses, 2006.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

View-dependent Triangle Mesh Simplification using GPU-accelerated Vertex Removal

Thomas Odaker
Ludwig-Maximilians-Universitaet
Muenchen, Germany
odaker@a1.net

Dieter Kranzlmueller
Ludwig-Maximilians-Universitaet
Muenchen, Germany
kranzlmueller@ifi.lmu.de

Jens Volkert
Johannes Kepler University
Linz, Austria
jv@ica.jku.at

ABSTRACT

We present an approach to view-dependent triangle mesh simplification based on vertex removal, which focuses on allowing the execution of a large number of operations in parallel. The individual vertex removal operations are designed to be applied without any need for communication or synchronisation between operations, thus allowing an efficient implementation on modern GPUs to reduce the computation time for the coarse mesh.
Since we cannot compute the entire simplification in a single step and have to perform several iterations of parallel vertex removal, we aim to maximize the number of vertices removed from the mesh in each iteration to efficiently use the available hardware and reduce the number of necessary iterations. The removal operation is based on the half edge collapse and avoids mesh foldovers and topological inconsistencies at each step.

Keywords

mesh simplification, level of detail, half edge collapse, computer graphics, view-dependent simplification, real-time rendering

1 INTRODUCTION

Simplification of triangle meshes is a commonly used approach to reduce geometric data and the performance necessary for processing a mesh. Ever since it was first introduced in [Cla76a], a wide variety of techniques and algorithms that compute a coarse mesh have been presented.

In [Oda15a] we introduced our approach to view-dependent simplification that is designed for an execution on a GPU. In this paper we introduce further developments and improvements to this approach that increase parallelism and quality of the simplification. This leads to a significant reduction of the number of necessary iterations and shorter processing times.

2 RELATED WORK

We classify simplification algorithms into those used in a preprocessing step and algorithms executed at runtime. Creating a coarse mesh in a preprocessing step eliminates the need for fast processing times and allows higher quality simplifications. Creating a coarse mesh in real-time before rendering each frame enables view-dependent simplification that minimizes visible artefacts for a given camera position, but calls for fast run times not to slow down the overall rendering process.

A variety of simplification operators have been defined, some of which have been adapted for real-time usage and even execution on modern GPUs to further speed up the simplification process. We now present the three operators most important to us:

Vertex clustering First presented in [Ros92a] this operator superimposes a number of cells over the volume of a mesh. All vertices within a cell are collapsed into a single vertex and the model data is updated accordingly. This approach can be used for fast processing times, but it may create low quality simplifications since the topology of the mesh is not preserved. In [DeC07a] a GPU-accelerated algorithm based on this operator designed for real time simplification is presented.

(Half) Edge collapse The edge collapse originally presented in [Hop93a] is the replacement of an edge and its two endpoints with a single vertex. The half edge collapse is a more restricted version that replaces the edge with one of its endpoints. The edge collapse is widely used. One common example for an algorithm relying on the edge collapse is progressive meshes ([Hop96a]). An improved version of this algorithm (modified for execution

on a GPU and real-time execution) is presented in [Hu09a] and [Hu10a].

Vertex removal This operator presented in [Sch92a] removes a vertex from the mesh and retriangulates the resulting hole. Different algorithms for retriangulation are available, one of them being the half edge collapse. For a vertex to be removed, one of the edges to a neighbouring vertex is chosen and a half edge collapse executed on it. The replacement position is the neighbouring vertex.

In [Oda15a] we presented our approach to triangle mesh simplification, with further discussion of details and results in [Oda15b]. All vertices of a given triangle mesh are analysed and classified as to be removed or to remain in the mesh. A parallel vertex removal technique based on half edge collapses is used to remove as many vertices as possible in parallel. Additionally we introduced a set of per-vertex boundaries that prevent mesh foldovers and topological inconsistencies despite the parallel execution.

In this paper we present further development and improvements to this algorithm that aim to increase parallelism, decrease runtime and improve the quality of the coarse mesh.

3 PREVIOUS WORK

This paper is based on our previous work presented in [Oda15a]. We further discussed details and results of this approach in [Oda15b]. In this section, we present an overview of our previous approach that we improved on.

The algorithm executes three steps: classification, parallel removal and reclassification (Fig. 1).

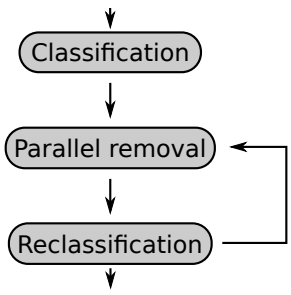


Figure 1: Algorithm steps

Classification This step analyses all vertices of a triangle mesh and marks them as to be removed or to remain. A simple metric that relies on the average distance between the tangential plane of a vertex and the neighbours is used to compute a vertex error, which is used for classification. This metric was primarily chosen for short computation times. Additionally, to prevent all vertices from being selected for removal, we introduced a layered error manipulation, artificially increasing the vertex error of some vertices to guarantee that they remain in the mesh. This leads to improved parallelism and guarantees the functionality of the algorithm.

Parallel removal We define any vertex that is selected for removal with at least one neighbour that is to remain as a removal candidate. A parallel removal step tries to remove all current removal candidates simultaneously, using a set of per-vertex boundaries to prevent foldovers and topological inconsistencies. The removal of vertices changes edges of the mesh which results in a new set of removal candidates being created.

Reclassification After each removal step, the classification of all vertices still selected for removal is updated to adapt to changes to the mesh and improve the quality of the coarse mesh.

Several iterations of parallel vertex removal and reclassification are executed. Due to how removal candidates are defined, only a part of vertices selected for removal can usually be processed in a single iteration. When a vertex is removed from the mesh, one or more edges are changed. If a vertex V is removed by a half edge collapse, any neighbour of V selected for removal becomes a removal candidate. This approach allows all vertices selected for removal to be processed over the course of several iterations.

We also addressed the issue of deadlocks. The per-vertex boundaries are designed to allow the removal of neighbouring vertices without any communication and can block valid combinations of half edge collapses. This potentially causes a situation, where multiple neighbouring vertices block each others' removal. Due to the parallel nature of the algorithm, it is not possible to identify deadlocks until the subsequent iteration. This can result in additional iterations being necessary to resolve deadlocks and delaying the completion of the simplification.

While our original approach worked well and resulted in fast processing times, we identified several shortcomings of the algorithm that limited parallelism and potentially the quality of the simplification. In this paper we present an improved algorithm that is based on our previous work and focuses on improving the classification and overall parallelism.

4 OVERVIEW

We identified several limiting issues with our original approach that we want to improve on:

Vertex error manipulation During the vertex classification we introduced a vertex error manipulation

based on points arranged in a grid to avoid all vertices being selected for removal. While this guaranteed that the algorithm could remain functional at all times, it ignored irregular triangle sizes and did not always suit the given mesh well.

Removal candidates We defined a removal candidate as a vertex selected for removal that has at least one neighbour that is to remain in the mesh. In each iteration, all removal candidates are processed. This results in a potentially large number of vertices selected for removal being ignored, since they currently have no neighbour that is to remain in the mesh, which reduces parallelism.

Replacement positions Only vertices selected to remain in the mesh are defined as valid replacement positions, potentially ignoring neighbours that would be better suited as a replacement position.

For the improved algorithm we rely on the vertex error we presented in [Oda15a]. The layered error manipulations used in our original approach helped to guarantee the functionality of the algorithm and improve the parallelism. In this paper we still rely on error manipulation to guarantee that some vertices are always selected to remain in the mesh. This is, however, not done using regularly spaced points in multiple layers. Instead a number of vertices is selected based on the minimum number of edges between them and their vertex error is set to a very high value to guarantee they are always selected to remain in the mesh.

To improve parallelism, we introduce the concept of auxiliary vertices (see subsection 5.1). For a mesh a set of auxiliary vertices is precomputed. An auxiliary vertex, that is not currently a removal candidate, can be used as a replacement position for neighbouring vertices, potentially greatly increasing the parallelism of the algorithm.

After computing the classification for all vertices of a mesh, the initial removal candidates are computed. We modify the definition of a removal candidate to include vertices selected for removal, that have at least one neighbouring auxiliary vertex, that is not currently a removal candidate. We execute the parallel removal step based on the half edge collapse using the per-vertex borders from [Oda15a] to prevent foldovers and topological inconsistencies.

The last step is the reclassification of vertices. It updates the vertex error of vertices selected for removal, that have not yet been removed. This step is used to adapt the classification to changes made in the previous parallel removal step and improve the quality of the simplification.

5 CLASSIFICATION

The classification step in [Oda15a] computes an error value for each vertex of a mesh. This error represents the difference between the mesh before and after a vertex removal operation. The original metric computes the error based on the geometric data and then scales it using the view vector and position of the camera. All vertices with a scaled error below a user-defined threshold are selected for removal. In our previous work we identified a problem with this approach: all vertices being selected for removal prevent the execution of our algorithm. Additionally a low number of vertices remaining in the mesh results in few removal candidates, which reduces parallelism. In order to avoid these problems we introduced a manipulation of the vertex error. A small number of vertices is assigned a very large vertex error to guarantee, that they always remain in the mesh.

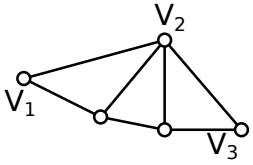


Figure 2: Minimum number of edges between vertices

The approach to selecting vertices for the error manipulation we propose in this paper is based on a minimum number of edges between two vertices with a manipulated error value. An example for the minimum number of edges between vertices is shown in Fig. 2. The vertices V_1 and V_2 have a minimum of 1 edge, while V_1 and V_3 have a minimum of 2 edges between them. The amount of vertices selected by this algorithm determines the maximum simplification and is controlled by the number N of edges. This value is chosen by the user. The number N can be used to influence the grade of the maximum simplification as well as the processing time: smaller N can reduce the number of necessary iterations for the simplification and improve processing times, while larger N allow the removal of additional vertices and can create a coarser mesh. Vertices are selected based on the maximum curvature of the surface to better maintain the shape of the object.

The first step to find vertices for error manipulation is to calculate the principal curvatures and store the maximum curvature of the surface in each vertex. We want to determine a set G of vertices that are guaranteed to remain in the mesh.

Initially, the vertex with the maximum curvature is selected and added to G ($G = \{g_0\}$). It is the first vertex selected for error manipulation and the starting point for the further steps of the algorithm. Given the maximum curvature for each vertex and the value N, the vertices are selected as follows:

1. Find all vertices $C = \{c_0, c_1, ..., c_n\}$ that have a minimum of N edges to any vertex in G and a maximum of N edges to at least one vertex in G.

2. Find the vertex c_i with the maximum curvature of all vertices in C.

3. Add c_i to G.

These three steps are repeated until no more vertices can be found in step 1. Then all vertices in G are assigned a very large error value to guarantee that they remain in the mesh.

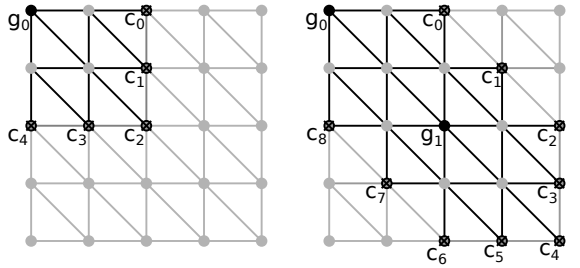


Figure 3: Vertex selection for error manipulation

Fig. 3 shows an example for this approach. On the left side the top left vertex (g_0) has been selected as the initial vertex in G and the list C has been filled in step 1 of the first iteration ($C = \{c_0, c_1, c_2, c_3, c_4\}$). For this example N has been set to 2. The right side illustrates the mesh after the first iteration is completed. The central vertex (g_1) has been selected and added to G. The set C has been updated in step 1 of the second iteration, resulting in a new set of candidates C for error manipulation.

The layered approach also has the goal to create additional removal candidates. Besides Layer 0 with vertices that always remain in the mesh, additional layers are created. Vertices selected in these have their vertex error increased to have additional vertices remain in the mesh at certain distances between the object and the camera. While this approach increases parallelism and reduces processing times, it can result in a simplified mesh, where the vertices are arranged in a grid like structure. Fig. 4 shows an example for a simplification of the Stanford Bunny illustrating this phenomenon. In this paper we replace the layered error manipulation with the concept of auxiliary vertices. In a preprocessing step a number of vertices with an unmodified vertex error is selected and marked as auxiliary vertices. During the simplification process any auxiliary vertex, that is not currently a removal candidate, can be used as a replacement position for neighbouring vertices marked for removal. Auxiliary vertices can be selected for removal. They are no longer considered auxiliary vertices and become removal candidates once they have a neighbour that is to remain in the mesh.

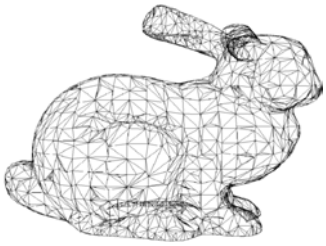


Figure 4: Simplified mesh with vertices arranged in a grid-like structure

After the vertices for error manipulation have been selected, we determine the auxiliary vertices.

5.1 Auxiliary vertex computation

Since auxiliary vertices can be replacement positions for neighbouring vertices selected for removal, we modify the definition of the removal candidates. In [Oda15a] any vertex selected for removal, that has at least one neighbour that is to remain in the mesh, is a candidate. In this paper, we include the auxiliary vertices in this definition so that a removal candidate is any vertex of a given mesh that:

- is selected for removal.
- has at least one neighbour, that is to remain in the mesh, or has at least one neighbour, that is an auxiliary vertex, which is not currently a removal candidate.

The idea of the auxiliary vertices is to make sure that as many vertices selected for removal as possible can be processed in any iteration. Since auxiliary vertices are determined in a preprocessing step and the classification is done at runtime, we do not know which vertices are to remain in the mesh when selecting the auxiliary vertices (with the exception of vertices in G).

For the purpose of computing auxiliary vertices, we assume that all vertices in G are to remain in the mesh and all other vertices are selected for removal. Given this assumption, we want all vertices of the mesh to be either vertices selected to remain in the mesh, auxiliary vertices or removal candidates. Based on this we compute auxiliary vertices as follows:

1. Find removal candidates First the list of removal candidates is determined, taking into account all vertices in G selected to remain in the mesh as well as the current list of auxiliary vertices (empty list initially).

2. Auxiliary vertex candidates The list of candidates $C = \{c_0, c_1, ...c_n\}$ for the auxiliary vertices is selected. It contains all vertices chosen for removal, that have a neighbouring removal candidate and are not currently a removal candidate or an auxiliary vertex.

3. Auxiliary vertex selection Selection of one of the candidates in C as auxiliary vertex. This is done using the candidate with the maximum curvature.

These steps are repeated until all vertices of the mesh not in G are either removal candidates or auxiliary vertices. This approach allows a greater number of removal candidates and therefore increases parallelism and prevents a large number of vertices selected for removal from being ignored during the parallel removal step.

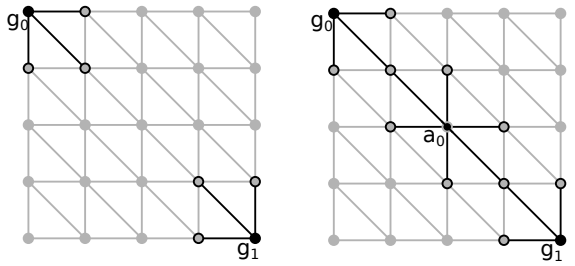


Figure 5: Auxiliary vertex selection

Fig. 5 shows an example of the selection of the auxiliary vertices. On the left side the vertices in the upper left (g_0) and bottom right corner (g_1) are in G. Their neighbours are assumed to be removal candidates. On the right side the center (a_0) vertex has been chosen as an auxiliary vertex. Additional removal candidates are available. The steps 1-3 are repeated until no more auxiliary vertices can be created.

During classification auxiliary vertices are treated as any other vertex and can be classified as to remain in the mesh or to be removed. If the vertex is selected to remain in the mesh, it is a valid replacement position for neighbouring removal candidates and no longer an auxiliary vertex. If it is selected for removal, it remains an auxiliary vertex until it has a neighbour that is selected to remain in the mesh and it becomes a removal candidate.

6 VERTEX REMOVAL

Compared to [Oda15a] the removal step remains mostly unchanged. All removal candidates are processed in parallel. For each candidate the possible replacement positions are determined, the per-vertex boundaries (which block any half edge collapse that may cause a mesh foldover or topological inconsistency) computed and one half edge collapse is executed.

We do, however, change the definition of possible replacement positions. Our original approach only used vertices that were selected to remain in the mesh. This was chosen to make sure that each vertex was moved to its final position and allow for an efficient implementation. With the introduction of auxiliary vertices a vertex can be moved to the position of a neighbour that has to be removed in a later iteration.

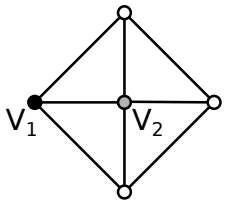


Figure 6: Possible replacement positions

We therefore change possible replacement positions for a vertex to include every neighbour that is not currently a removal candidate. This definition differs from that of the removal candidates, since it also includes vertices, that are selected for removal but are neither auxiliary vertices nor removal candidates. Fig. 6 shows an example. V_1 is to remain in the mesh, V_2 is a removal candidate. The remaining vertices are selected for removal but currently not removal candidates. In our previous approach, only V_1 is considered a possible replacement position, while the improved algorithm can move V_2 to any of its neighbouring vertices. The definition of a removal candidate is chosen to guarantee that each vertex that is processed has at least one possible replacement position and to avoid vertices that cannot be removed. At the same time we want to allow the most amount of freedom when choosing a replacement position.

In [Oda15a] we discuss the problem that our per-vertex borders can block combinations of half edge collapses when applied to neighbouring vertices. This can lead to multiple neighbouring vertices blocking each others' removal, causing a deadlock and delaying completion of the simplification process. Allowing any neighbours that are not removal candidates to be chosen as a replacement position has the potential to avoid blocked replacement positions, reduce deadlocks and improve the quality of the coarse mesh.

7 RESULTS

We devised an implementation of our improved algorithm using Nvidia CUDA and ran multiple tests on a Geforce GTX 670 GPU with 1 344 cores. Several models from the Stanford 3d Scanning Repository (Stanford Bunny, Armadillo, Dragon and Happy Buddha) were chosen for testing purposes.

Fig. 7 shows simplifications of the Stanford Bunny and Armadillo and compares them to the original meshes (from left to right: Stanford Bunny original, Stanford Bunny simplified, Armadillo original, Armadillo simplified). Fig. 8 shows the same comparison for Dragon and Happy Buddha. For these examples about 90% of the triangles of the original meshes have been removed. The models vary in terms of vertex and triangle count and were chosen to analyse how the improved algorithm scales with an increasing number of vertices.

Table 1 shows the number of vertices and triangles the models we used for testing contain.

Figure 7: Comparison: Original (left) and simplified (right) mesh for Stanford Bunny and Armadillo

Figure 8: Comparison: Original (left) and simplified (right) mesh for Dragon and Happy Buddha

Model	Vertices	Triangles
Stanford Bunny	35 947	69 451
Armadillo	172 974	345 944
Dragon	437 645	871 414
Happy Buddha	543 652	1 087 716

Table 1: Number of vertices and triangles in the models used for testing

For testing purposes a simplification was computed for each model that removed a majority of the triangles of the original mesh. The most important factor to us is the overall runtime of the simplification process. As described earlier, the maximum simplification is determined by the number N for the vertex error manipulation. We chose this number separately for each model to incorporate its vertex count.

Model	Triangles rem.	N	Time (ms)
Stanford Bunny	62 100	4	5.2
Armadillo	324 164	6	26.1
Dragon	821 161	7	52.6
Happy Buddha	1 027 314	7	66.5

Table 2: Triangles removed, number N and processing time for the simplification in milliseconds for each model

Table 2 shows an overview of the results of the simplification process. For each model, the number of triangles removed, the number N chosen and the processing time in milliseconds are listed. The Stanford Bunny - being the model with the fewest vertices and triangles - was simplified in only 5.2 ms with a triangle reduction of about 90%. The triangle count of the largest model in these tests (Happy Buddha) was reduced by about 94% with a runtime of less than 67 ms.

In addition to the overall processing time we measured the number of necessary iterations.

Model	Iterations
Stanford Bunny	7
Armadillo	15
Dragon	18
Happy Buddha	20

Table 3: Number of iterations necessary to complete the simplification for each model

Table 3 shows the necessary iterations for all models. The simplification process for the Stanford Bunny took 7 iterations while the coarse mesh for Happy Buddha was created in 20 iterations. In addition to the number of iterations, we measured the number of removal candidates in each iteration for the Stanford Bunny.

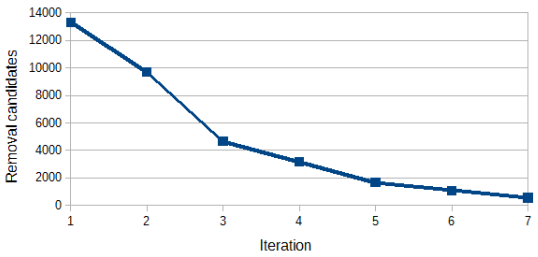


Figure 9: Removal candidates per iteration for the Stanford Bunny

Fig. 9 shows the number of removal candidates in each iteration for the simplification of the Stanford Bunny. This graph does not show how many vertices were actually removed in each iteration. Some vertices may not have a valid replacement position due to the per-vertex borders and cannot be removed. They remain in the mesh and are again removal candidates in later iterations. In the first iteration 13 321 removal candidates are available for processing. This number drops throughout the process with 1 089 and 555 removal can-

didates in the last two iterations. While the later iterations cannot fully utilize all available cores of the GPU (1 344), the majority of the iterations offers more removal candidates than cores.

In addition to the performance measurements we compared the results to those of our previous algorithm.

7.1 Comparison to previous work

In [Oda15b] we presented the results of our previous work and compared it to existing algorithms. In this section we will compare the results of the improved algorithm to those in [Oda15b]. Overall we expected to see the highest performance gain when simplifying the models Dragon and Happy Buddha due to the higher vertex count.

For the performance measurements in this paper we chose simplifications of the models that result in a similar number of vertices/triangles to the ones in [Oda15b]. This allows us to directly compare the results of the two algorithms and easily determine any performance gains. Since the algorithm in this paper is designed to improve parallelism and reduce the number of iterations, we expect to see improved run times and a reduced number of iterations compared to previous results. At first we compare the overall runtimes.

Model	Impr. alg.	[Oda15b]
Stanford Bunny	5.2 ms	5.7 ms
Armadillo	26.1 ms	29.1 ms
Dragon	52.6 ms	80.1 ms
Happy Buddha	66.5 ms	96.1 ms

Table 4: Comparison of runtimes between the improved algorithm and the results in [Oda15b]

Table 4 shows the runtime comparison for all four models. The reduction of runtime for the models Stanford Bunny and Armadillo is about 9%, while the improved algorithm can greatly reduce the processing time for models with a larger number of triangles (greater 30%). This comparison shows that the new algorithm can greatly reduce the runtime of the simplification.

The second measurement we showed earlier is the number of iterations and removal candidates in each iteration for the Stanford Bunny. Our new algorithm computes the coarse mesh of the Stanford Bunny in 7 iterations, while the results in [Oda15b] show that 12 iterations were necessary. The simplification of Happy Buddha took 33 iterations using our previous approach, while the improved algorithm finished after 20 iterations.

Fig. 10 shows the comparison of the number of removal candidates in each iteration between our original and improved algorithm for the Stanford Bunny. The 12 iterations for our original algorithm processed between 7 436 and 74 vertices with a very low number of

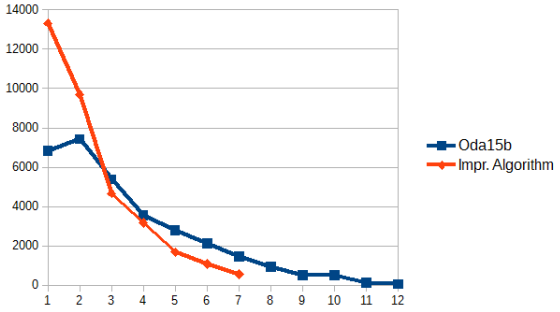


Figure 10: Comparison: Removal candidates per iteration for the Stanford Bunny

removal candidates in the last iteration (74 compared to 555 removal candidates for the improved algorithm). The modified vertex error manipulation and the introduction of auxiliary vertices improve parallelism and allow a better utilization of the hardware.

8 FUTURE WORK

Future work can improve the classification and reclassification steps. As the classification is currently based on a geometric error value and does not take the removal of neighbouring vertices into account, an unnecessary large number of vertices can be selected for removal. All these vertices need to be processed and may be reclassified. An improved error metric and better selection of vertices for removal can reduce the number of vertices selected for removal, reduce the number of vertices that need to be processed and therefore lead to an increase in performance. Greatly improving the classification may even render the reclassification step obsolete. This can lead to a better quality of the coarse mesh as well as shorter processing times.

9 CONCLUSION

The improvements made to our original algorithm have shown the potential to significantly reduce the runtime of the simplification by increasing the parallelism and reducing the number of necessary iterations. Especially when using models with a large number of vertices and triangles the improvements lead to a reduction in processing times of over 30%. The increased parallelism allows us to better utilize the parallel processing power of modern GPUs. The modified vertex error manipulation is better suited for models with irregular triangle sizes that proved to be disadvantageous to our previous approach as it could lead to a higher number of necessary iterations. Additionally the introduction of auxiliary vertices helps to reduce the number of vertices that cannot be processed in an iteration and improve parallelism.

On the other hand our algorithm still relies on error metrics designed for speed, fast update times and the isolated execution of the vertex removal operations. These

factors may lead to a decrease in overall quality of the coarse mesh and be limiting factors for the simplification.

10 REFERENCES

[Cla76a] Clark, J. H. Hierarchical geometric models for visible surface algorithms, Com. of ACM 19, No. 10, pp.547-554, 1976

[DeC07a] DeCoro, C., and Tatarchuk, N. Real-time mesh simplification using the GPU, I3D 2007 Proceedings of the 2007 Symposium on Interactive 3D Graphics Vol. 2007, pp.161-166, 2007

[Hop93a] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., A., and Stuetzle, W. Mesh optimization, ACM SIGGRAPH Proceedings 1993, pp.19-26, 1993

[Hop96a] Hoppe, H. Progressive meshes, ACM SIGGRAPH 1996 Proceedings, pp.99-108, 1996

[Hu09a] Hu, L., Sander, P., V., and Hoppe, H. Parallel view-dependent refinement of progressive meshes, I3D 2009 Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, pp.169-176, 2009

[Hu10a] Hu, L., Sander, P., and Hoppe, H. Parallel view-dependent level of detail control, IEEE Transactions on Visualization and Computer Graphics Vol. 16, No. 5, pp.718-728, 2010

[Oda15a] Odaker, T., Kranzlmueller, D., and Volkert, J. View-dependent Simplification using Parallel Half Edge Collapses, Proceedings of WSCG 2015, pp.63-72, 2015

[Oda15b] Odaker, T., Kranzlmueller, D., and Volkert, J. GPU-Accelerated Real-Time Mesh Simplification Using Parallel Half Edge Collapses, Mathematical and Engineering Methods in Computer Science: 10th International Doctoral Workshop, MEMICS 2015, pp.107-118, 2016

[Ros92a] Rossignac, J., and Borrell, P. Multi-resolution 3D Approximations for Rendering Complex Scenes, Modeling of Computer Graphics: Methods and Applications, pp.455-465, 1992

[Sch92a] Schroeder, W., J., Zarge, J., A., and Lorensen, W., E. Decimation of triangle meshes, ACM SIGGRAPH Computer Graphics Vol. 26, No. 2, pp.65-70, 1992

Enabling Gesture Interaction with 3D Point Cloud

Harrison Cook
School of Computing, Engineering and Mathematics, Western Sydney University
harrison.cook42@gmail.com

Quang Vinh Nguyen
MARCS Institute and School of Computing, Engineering and Mathematics, Western Sydney University
q.nguyen@westernsydney.edu.au

Simeon Simoff
MARCS Institute and School of Computing, Engineering and Mathematics, Western Sydney University
s.simoff@westernsydney.edu.au

Mao Lin Huang
School of Software, Faculty of Engineering & IT, University of Technology, Sydney
Mao.Huang@uts.edu.au

ABSTRACT

This paper presents a novel 3D point cloud gesture recognition system, based on an existing low-cost, accurate and easy to implement 2D point cloud gesture recognition system called $P. Our work improves recognition rates and lowers algorithmic complexity. We develop new 3D gestures, such as the GUN gesture and the SHAKE gesture, while also developing 3D poses like the L pose, OK pose, ROCK pose and PEACE pose for the LeapMotion Device. We demonstrate proposed gesture and pose methods on various 3D environments including a Monsoon mini-game, a cave painting interaction and a target practice scene. The average recognition rates for 3D gestures and poses were compared against the 2D, 3D and 3D+ recognition systems. The results indicate that most gestures in the proposed system were improved in comparison to the existing ones.

Keywords

User Interaction, Gesture Recognition, Finger Interaction, Leap Motion, 3D Point Cloud.

1. INTRODUCTION

Gesture recognition refers to determining when a gesture has occurred and to the general process of determining when a gesture has started and stopped [Yin14]. Natural gestures can be grouped into manipulative and communicative gestures. Manipulative gestures are about moving or interacting with objects, such as pressing a button or rotating an object around, while communicative gestures have the intent of conveying information to others. Communicative gestures can be an interpretation or movement via a symbol or via an act. A gesture via a symbol is often conveyed with a static hand pose and a gesture via an act is determined by the movement of the hand itself.

While natural gestures work in the real world, determining when the user is making a gesture requires our computer implementation to take on a more structured approach of flow and form gestures. A flow gesture could be a continuous gesture, where it progresses over a period of time or series of moments of time. The form gesture can be defined by determining whether the gesture follows a distinct path (such as scrolling a webpage based on position of words) or if it is based on a pose. This research uses the flow and form gestures as a guide to develop our own discrete and continuous gestures to allow the user a range of possibilities.

Hand gesture interaction should be simple enough to understand, while distinct enough so that the user understands which gesture has occurred. For example, uWave is an interactive application using gestures for various mobile devices that have a very small custom gesture recognition system to allow maximum space on the device [Liu09]. We expand this idea for maximum recognition rates while using only a small dataset.

This paper presents gesture recognition techniques that aim to allow better recognition of 2D and 3D gestures by extrapolating gesture sets using similar classifiers. We develop new gesture recognition algorithms that provide a seamless and effective natural interaction at various distances and screen resolutions. The contribution of our paper is as follows.

- New gesture recognition algorithms for fingertip interactions. The gesture pattern recognitions are identified in a simpler and faster way than the available techniques that use databases to store gesture templates for matching, such as [Ren11].
- An expansion of a 2D gesture recognition system [Vat12] into 3D, based on utilising LeapMotion device as input for the gesture recognition system.
- An evaluation of the developed system in terms of user experience and its effectiveness.

2. RELATED WORK

Kinect devices have been useful for medical practices for doctors and staff to interact with patient information without the requirement of the typical mouse and keyboard and the more inaccurate voice recognition [Ebe13]. Using third party libraries such as OpenNI [Pri16] and Libfreenect [Ope16] they were able to implement finger gestures into the OsiriX system [Ebe13]. This was achieved by taking the Kinect depth image, collecting the closest objects to the sensor then approximating the hand and fingers based on this blob. Once the hand and fingers have been recognised from the blob, each point around the hand is recorded and over time is checked with a database of other hands already recorded. However, the use of the Kinect V1 in [Ebe13] restricts the range of the interaction due to the limitations of the depth camera on these devices. The other major limitation of their system is the lack of explanation of the gestures to the user.

Another major work develops a new Kinect hand recognition system using the 'golden' energy function [Sha15]. It renders all possible poses of the hand and selects the pose, whose corresponding rendering best matches the input image while accounting for the prior probability of poses. The technique uses regions of interest (RoI) to determine the approximate hand position from the depth image stream and a learned pixel-wise classifier [Sho13]. When used with 3Gear [3Ge16] and LeapMotion device, this technique achieves a better percentage of 3D hand detection than those described previously. This system requires extremely large FingerPaint datasets of 3.2GB of storage space [Sha15] that could make portability and memory management an issue. High memory latency and a high end GPU are required when accessing the dataset in the memory in real time. In addition, the inability to work with both hands and the lack of a gesture recognition system for fingers could limit the utilisation.

Tang [Tan11] introduced a method for identifying grasping and pointing gestures in which a person's hand model was estimated based on a skeletal tracker. This is limited in terms of tracking resolution and inability to track at a close distance. Chen and Wong introduced an interactive sand art drawing using Kinect [Che14]. Four key gestures were detected by the system, including sing-point finger for single point erosion, single splicing and leaking, V-shape 2 fingers for pinch spilling and pinch erosion. Lee and Tanaka used databases for fingertips and hand shapes to enable gesture cognition with any finger in the palm as well as two fingers (thumb and index finger) [Lee13]. The interaction techniques were applied to sample applications for finger painting and mouse controlling. Song et al proposed an algorithm for gesture recognition algorithm based on depth information from a Kinect device [Son13]. Cook et al also introduced a real-time finger-gesture interaction system using Kinect v2 that identifies finger gestures at close range [Coo15]. However, vision-based systems are often limited in terms of accuracy, occlusion and inconsistency when changing the environment, such as dark light and rough background.

3. GESTURE RECOGNITION

Our gesture recognition system was developed based on gestures as point clouds approach [Vat12], to address the problems identified in Section 2. The Point-Cloud Recogniser ($P) is a gesture recognition system designed for quick and simple recognition of two dimensional gestures from a user's input. $P works by collecting a bunch of pre-determined points already created in the system, i.e. the database of gestures that the system can recognise. When ready to, it takes the users current points as an input to determine the closest matching set of user points to data points using a nearest neighbour classifier.

We expanded the recognition system from 2D to 3D that improved recognition accuracy and performance. The enhancement includes additional classifiers to the dataset to improve comparisons and collections of points by segmenting cloud stroke points based on certain parts of the hand and fingertips.

Gesture Recognition Systems

Gesture recognition systems that use $P include single stroke ($1) and multi stroke ($N) systems. A single stroke system can handle only one stroke on a canvas for comparisons. A multi stroke system can handle more than one stroke. $P uses the $1s algorithm for nearest neighbour matching and gesture recognition, while also using $Ns algorithm for allowing multiple strokes to be recognised. The benefit of the dollar family gestures is how users can add their own variations to the data set to be used for future recognition [Wob07].

The uni-stroke recogniser ($1) developed by [Wob07] is the first gesture recognition system developed in the $ family. Its implementation is similar to the $P system, which uses the nearest neighbour and Euclidean scoring systems. $1 is an extension of an existing recognition system called SHARK^2 [Kri04]. It performs extremely well with recognising gestures even when given a low amount of training samples to compare user input.

The multi-stroke recogniser ($N) is a solution to $1's problem of not being able to use multiple strokes to determine a gesture [Ant10]. $N was designed so that the input of strokes would be stroke and direction invariant. This was achieved by recording all distinct directional behaviours of the gesture to compare the user data. $N has an immensely increased complexity by loading all of these permutations. This issue could

be improved by removing multiple stroke types and directional changes when comparing single stroke gestures [Ant10]. Protractor uses a closed-form template-matching method eliminating the simplicity and complexity of the Golden Section Search algorithm [Ant12].

We adopted multi-stroke recognition system that could handle multiple start and stop points. The system needs to be fast without the need to remember any permutations in the strokes as the stroke order is not important for our gesture recognition. This rules out $N and $N-Protractors complex algorithms leaving $P. $P does have one drawback with its implementation, where the point clouds are rotation variant. This means that each gesture is not rotated when comparing point clouds. To overcome this problem, when we developed the database we added gestures of different angles to the training set to compensate for this drawback.

$P in a 3D world space ($P3D)

Extending $P into a 3D implementation was done by following the same algorithms in 3D space, such as Protractor [Kra11]. We utilise the practicality, simplicity and usability of the $P algorithm, extending it to operate with three-dimensional gestures.

We take the existing method for gathering and assigning points with each gesture created and change the creation method depending on the data. Because the protractor recogniser takes the Euclidean distance between points in 2D to determine the distance, they take the Z position of the set of points when applicable. The result means that the distances between points when classifying gestures will be slightly more between 2D and 3D gestures because of the added axis.

The 3D centroid formula determines the centre of the 3D data points which is defined by equation 1. We used a 32-point sampling resolution (N) [Kra11] to define the insignificant change in accuracy. Translating the data requires the 3D centroid position to be equal to the subtraction of each of the data points with the centroid. The new translated data points will make the centroid be at position (0, 0, 0). This translating makes it possible to centre the data for recognition.

$$Cx = \sum_{i=0}^{N-1} x_i \Big/ N \,,\; Cy = \sum_{i=0}^{N-1} y_i \Big/ N \,,\; Cz = \sum_{i=0}^{N-1} z_i \Big/ N \qquad (1)$$

Where Centroid $C = (Cx, Cy, Cz)$ and N is the sampling resolution.

Resampling the data points is the most crucial of the pre-processing techniques. It requires multiple calculations to convert the data from any number of points to the sampling resolution of 32 that we want for 1:1 conversions between user data and gesture data. We extended the 2-dimensional method to create the linear interpolation (LERP) point as equation 2, where a new 3D point (P) was calculated by LERP point (∂), first point (f) and current point (p).

$$\alpha = \frac{(I-D)}{D} \quad where\ D\ != 0$$
$$\partial = \begin{cases} \partial & if \quad \alpha > 0.0 \quad and \quad \alpha < 1.0 \\ 1.0 & if \quad \alpha \geq 1.0 \\ 0.0 & if \quad \alpha \leq 0.0 \end{cases}$$
$$P.x = (1.0 - \partial) * f.x + \partial * p.x$$
$$P.y = (1.0 - \partial) * f.y + \partial * p.y$$
$$P.z = (1.0 - \partial) * f.z + \partial * p.z \qquad (2)$$

$P3D+

While $P3D generally provides good gesture recognition in 3D, its effectiveness normally depends on datasets and user inputs. We added to the $P system a set of classifications and ordering the LeapMotion data so that each finger and palm position was its own stroke and followed its previous position. Our goal is to improve the 3D recognition system by eliminating the need to search all the gestures within the database. We also improved pre-processing gestures by assigning stroke IDs to point clouds to remove inaccuracies. The enhancements are as follows.

Removing Redundant Point Clouds

We limited the complexity of the gesture recognition by only selecting gestures that reach a certain classification. These classifications would remove the ambiguity of searching through all possible gestures and instead focus on the range. We developed three classifiers in the $P3D system. The first classifier determines whether the recorded gesture was in 3D or 2D. This classifier only worked when there was a 2D gesture being recognised. The second classifier is for identifying whether the gesture is a pose or not. A pose gesture features the input to contain very little movement to no movement whereas a normal gesture requires much more movement over a series of time. The third classifier determines which left or right hand that the gesture is used with. This implementation aims to remove ambiguity and noise when comparing gestures from the left hand that could increase recognition when using the right hand. For example, a left PEACE pose matches well with a right OK pose and vice versa.

Assigning and Ordering Stroke IDs to the Hand

$P was designed for a 2D drawing scenario that was inputted one stroke at a time and it did not take the stroke ID into consideration. While this is normally fine when the user draws the gesture one stroke at a time, it could be a problematic when recording

different strokes currently. This is because the system thinks the entire gesture is made of single strokes.

$P does not take into consideration the stroke order, stroke direction and stroke permutations as in $N/$N-Protractor solutions [Ant10, Kra11]. It implements the stroke system for pre-processing, resampling and calculating the path length for the gesture. We assigned the palm and the fingers with their own stroke IDs. The IDs of *Palm*, *Thumb*, *Index finger, Middle finger, Ring finger*, and *Little finger* were numbered 0 to 5 respectively. The order of stroke indexes of the hand and fingers was used to improve the matching rate between these point clouds. This improvement provides a closer approximation on how to interpret the data where inaccuracies can be reduced to produce a closer match to determine the correct gesture.

4. INTERACTION WITH LEAPMOTION – A CASE STUDY

The LeapMotion is a controllable device that tracks hand and finger movements using LEDs and infrared cameras. While the Kinect can track skeletons from long distances with its sensors, the LeapMotion is dedicated to tracking hands and fingertips with a shorter range and more accurate than the Kinect.

Capturing the 3D Point Clouds

In order for the LeapMotion to be recognised by the point cloud gesture recognition system, we created the training set that was used in the classifications of the gestures. The chosen data from the LeapMotion system to determining these gestures were the stabilised palm position and the stabilised tip position of all the fingertips. Other points such as the wrist and distal bone positions were not considered for the training set because these details were not used for the recognition system.

Environment Building

We used Unity 3D game engine to develop our interaction environment. Unity provides an excellent framework for developing tools and applications within its integrated development environment (IDE). Unity's IDE allows for 2D or 3D applications with a variety of different tools that will improve a user's experience with the system.

It is crucial for first time users to learn about the system interactively. We developed a tutorial system to illustrate all the different elements of our environment in such a way that is easy to follow and understand. The system also allows the user to revisit old lessons when required. Once completing the tutorial, we provide a showcase scene where users can collaborate and manipulate objects within the scene using these new abilities learnt.

Monsoon Minigame

We developed a simple minigame to help the users be familiar with the interaction using the LeapMotion. The game revolves around the user trying to catch as many boxes, barrels, orbs and traffic cones as they can. The objects are spawned above the user and trickle down every second. It is up to the user whether they want to hold these objects or to play around with them (See Figure 1). This minigame serves as an initial learning task for beginners and a challenge for the experts who want the highest score.

Figure 1. Mosoon Minigame monsoon state.

Tutorial System

The tutorial system was developed to help the users understand the systems. They included 2D drawing, 2D gesture recognition, moving the camera, 3D pose recognition and 3D gesture recognition. Each of these systems is also a state within the tutorial system that the user can cycle through. Each state has its own visual, written and interactive way of showing the user the current state.

2D Drawing

This state trains the user how to draw 2D objects on a canvas. The drawing system requires the use of the LeapMotion and a set of criteria needs to be met to begin drawing (see Figure 2). To begin with, the only hand that the drawing system recognises is the primary hand selected by the user in the first tutorial state. The index finger is chosen for drawing. When the user wants to stop drawing, they can either move all their fingers back from the sensor, or open their thumb out without the need of moving. If a user makes a mistake and wishes to undo the previous stroke on the canvas, they can do so by swiping right.

2D Gesture Recognition

The 2D gesture recognition scenes goal is to help the user with recognising 2D shapes. While this process could have been combined with the drawing state, our experiences show that learning to draw first with the added gesture recognition was too steep a learning curve for first time users. Once users are

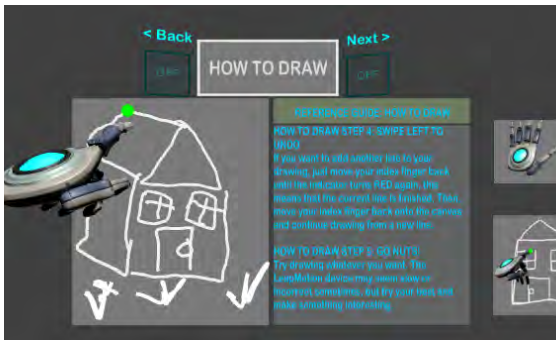


Figure 2. Tutorial state default template.

familiar with the drawn shapes, they can move their hand by positioning all the fingers forward on the LeapMotion device. The system then classifies the gesture using $P and returns the best scoring point cloud for each gesture. The implemented gestures in this tutorial stage are *Circle*, *Rectangle*, *Triangle* and *Cross*. The user can create shapes and then grab, move or throw them away.

Moving the Camera

This tutorial follows the main principle where users can move around the play area with their secondary hand by closing it into a fist. This scene is used as a breaker from the 3D gesture recognition.

3D Pose Recognition

The 3D pose recognition scene teaches users how to hold their secondary hand when performing a pose. To perform a pose, the user positions their hand so that it forms a symbolic gesture and hold that pose until it is recognised. The Pose recognition system can use 4 different poses including the L Pose, the ROCK Pose, the OK Pose and the PEACE Pose (See Figure 3).

Poses are recognised within the gesture recognition system through the use of velocity within the

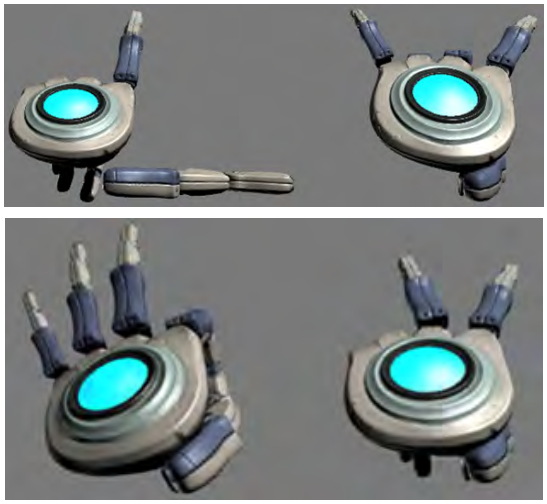


Figure 3. L Pose, ROCK Pose, OK Pose and PEACE Pose respectively.

LeapMotion device. We use this data to find the minimum x, y or z value from the hands and fingers velocity data. We then check if the hand has stopped moving with a velocity of under a minimum value. Once this occurs, the gesture recognition system records the position information from the fingers and palm for that frame. This process repeats until either the hand is moving quicker than the minimum velocity and our data points need to be reset, or the user holds their hand over the minimum number of frames required, 75 frames in our implementation.

Making users wait while performing a pose could be an issue if there is no hint or indication. To overcome this, a pose gesture indicator was presented showing the progression of a pose gesture until the pose recognition system can determine the pose.

3D Gesture Recognition

The 3D gesture recognition follows the 3D pose recognition state where we want to teach the user how to do gestures in the environment. To perform gestures, the user uses their secondary hand and makes a quick movement. As opposed to poses that require little to no movement, gestures requires a sudden movement that when slowed down is recognised as a gesture. We implemented two sample gesture recognitions in our system including SHAKE and GUN.

To recognise a gesture, the system gathers the same velocity data from the palm and fingers as with the pose recognition system but determines the maximum velocity from the x, y and z values instead. It then waits until the velocity from the palm or fingers is greater than the maximum velocity of 500 millimetres and then it starts recording the gesture. Once this happens, each frame is recorded until the hand or fingers velocity has fallen under 500 millimetres or the number of gesture frames is over our sampling resolution of 32 points. A gesture indicator was also created to assist the interaction.

Practice Systems

We developed two practice systems to complement the lessons taught in the LeapMotion tutorial environment. The first system is a target practice scene that uses 2D drawings to create objects for the user to fling around and 3D poses for altering those shapes properties. This scene also uses 3D gestures for users to interact with the environment as opposed to just the shapes. The second system is a *Cave Painting* system that emulates how users could use the 2D recognition system in a creative and fun way with an image matching game.

Target Practice System

The target practice system is the showcase scene for our LeapMotion environment. It contains a number of gestures in both 2D and 3D and offers an open

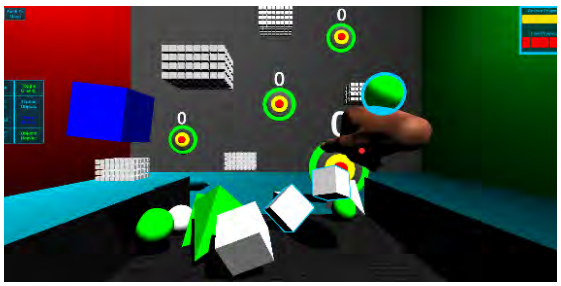


Figure 4. Target Practice system with multiple objects in play.

sandbox where the user can use all these different systems together to interact with the environment. The interactable objects within the play area are cube walls and targets. Targets generate points based on how close an object gets to the centre from the users throw and a cube wall acts as a breakable barrier which users can break open by using multiple objects at once (see Figure 4).

The types of 3D Poses that our system can recognise are: L Pose this turns the gravity off for all created objects, OK Pose changes the colours of all newly created objects, ROCK Pose triples the current objects created and the PEACE Pose makes all the objects heavier/lighter. The 3D gestures that the Target Practice scene can also recognise are SHAKE Gesture that shakes the camera around and the GUN Gesture that creates a bullet launched from the fingertips that explode on impact.

Cave Painting

The Cave Paintings goal is to closely match one of the 5 popular animals from the Australian outback (see Figure 5). The Bat and Turtle contain the fewest shapes; the Emu and Lizard are more complicated due to their curves and the Kangaroo is the most difficult one with both curves and difficult shapes. Users are required to trace the animal drawing using their finger as close as they can. The system gives the users feedback on how close they were to the drawing as well as acknowledge their performance. These messages do not have negative comments written on the drawing to provide an enthusiastic approach when trying to draw the animal.

5. EVALUATION

We compared the results of three different point cloud gesture recognition systems ($P, $P3D and $P3D+). Using the training set defined and created by using the program, we evaluated the performance (time complexity and the average recognition percentages) of the original 2D algorithm ($P), the added 3D gesture recognition algorithm ($P3D) and our improved 3D recognition algorithm ($P3D+). The gesture set consists of 15 gestures ranging from iconic aboriginal animals, 2D shapes, 3D poses and

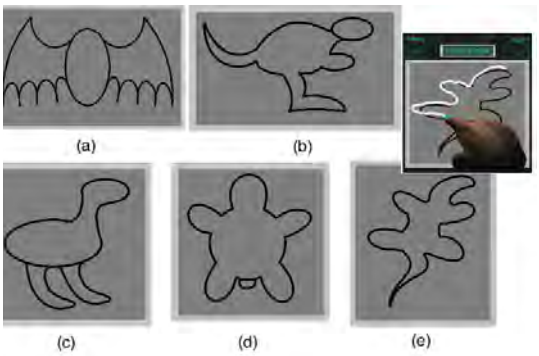


Figure 5. The 2D drawing gestures recognised in the Cave Painting scene. (a) Bat, (b) Kangaroo, (c) Emu, (d) Turtle, (e) Lizard.

3D gestures. The results in these experiments were evaluated based two sample gesture databases.

The first database consists of all 15 gestures from the gesture set, from which each pose per hand has 4 angle variations that were split into normal, rotated forward, rotated left/right (based on what hand was used) and rotated back. Each gesture on each left- and right-hand has five variations. Each of the 2D animal gesture contains three similar variations while each 2D shape also has five variations in average per shape. Overall, this database totals to 28 poses and gestures recorded (300kb) and 35 animal and shape drawings (600kb).

The second database expands the 3D gesture set by doubling the number of variations per hand to 10 each. The 3D poses also receive an increased number of variations per hand, from five to six. We also exclude all 2D gestures to focus on 3D recognition. To create unbiased results, the classification of the results uses the 3D poses and gestures from one database as the gesture/pose input to compare point clouds and determine the minimum distance by using another training set.

We initially used the scoring equation defined in [Wob07, Ant10]. However, this scoring method produces scores within the 95%-99% margin with very few scores ranging outside this segment. Our implementation requires the scores to range based on our observed minimum distances for the correct and incorrect gestures. To overcome this limitation, the scores were calculated using a polynomial formula to extrapolate the data explained above into a curve to map the distance data to the following percentage values, particularly *0.5 = 99%, 0.9 = 90%, 1.1 = 85%, 1.3 = 80%, 1.5 = 75%, 3.5 = 20%, and 4 = 0%.*

We do not consider the average recognition rate for $P on 2D gestures as the recognition average has already been conducted by [Ant12]. We test $Ps ability to recognise 3D gestures versus $P3D recognition system using both databases.

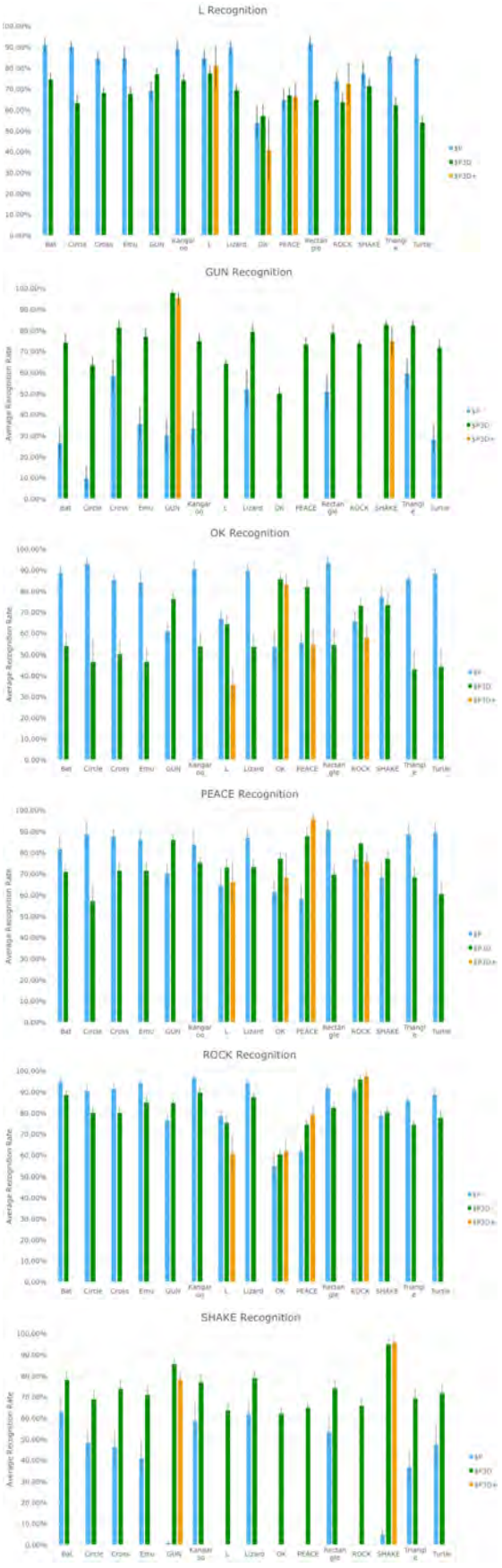


Figure 6. Gesture recognition averages using $P, $P3D and $P3D+ on small dataset with L, Ok, Gun, Peace, Rock and Shake Respectively.

First Dataset

From the results on the first dataset (see Figure 6), we can see a clear indication that $P could not determine any differences between 3D and 2D data. It is a surprise that the recognition rate in the L pose, $P proved to be more efficient when compared to $P3D (84% Range CI = [81, 87] to 77% CI = [74, 80]). However, the recognition rates for other gestures were higher in $P3D in comparison to $P particularly the L pose. There is no clear indication from the noise of the other gestures like Bat 90% CI = [87, 93], Circle 89% CI = [87, 91], Kangaroo 89% CI = [85, 93], Lizard 89% CI = [86, 92], Rectangle 91% CI = [88, 94] that the L pose was the most recognised gesture. All other poses and gestures have significantly decreased average recognition rates for the correct gesture with high recognition rates for the wrong gestures. With the ROCK pose, $P does have a high average recognition rate of 91% CI = [86, 96], but is still beaten by $P3Ds average of 95% CI = [93, 97]. This indicates our 3D implementation works better than the traditional 2D system.

Second Dataset

From the second dataset (see Figure 7) with a larger 3D database, $P performs miserably when determining the GUN gesture with an average recognition rate (< 50%) as well as an extremely low average recognition rate for SHAKE gesture (< 5% recognition mark). $P3Ds performs much better above the 95% recognition average, for example the SHAKE Left Hand (96% CI = [94, 98]), Right Hand (98% CI = [97, 99]) and GUN Left Hand (98% CI = [97, 99]) and GUN Right Hand (100%). When presented with a larger L pose database, the $P recognition average actually performs better than $P3D with the Left Hand scoring an excellent 90% (CI = [89, 91]) versus $P3Ds 85% (CI = [83, 87]) recognition average and the Right Hand achieving similar results with a 83% (CI = [76, 90]) average compared to $P3Ds 78% (CI = [72, 84]). This means that while the $P system performs worse on all gestures it has the ability to perform well or better than $P3D on pose recognition.

Comparing $P3D to $P3D+

With the improvements made to $P3D+, we evaluate whether our new $P3D+ performed better than the original $P3D. These evaluations follow the above comparisons, and T-value tests are used to determine if the improvements are statistically significant (alpha level of .05 for all statistical tests).

First Dataset

Figure 6 shows the GUN gesture has a decreased recognition rate from $P3Ds 98% (CI = [97, 99]) to $P3D+s 95% (CI = [92, 98]) but there is no statistical significant difference between these two systems

$t(38)=1.71$, $p=0.09$. This result means that while \$P3D+ recognition average is lower than \$P3D, there is not enough evidence to class both sets of data as different. With the PEACE Pose, we found that there was a significant difference in recognition averages for \$P3D 88% (CI = [84, 92]) and \$P3D+ 96% (CI = [93, 99]) $t(22)=3.01$, $p<0.01$. This result suggests that the improvements made to increase the recognition for the PEACE pose have increased the average enough to be statistically significant.

Second Dataset

Figure 7 also indicates that our hypothesis of the best-case scenario is apparent in the GUN (Left/Right) hand. For the Left Handed GUN gesture, the \$P3D Right Handed GUN recognition average is almost the same as the Left Handed average (99% CI = [98, 100]) while the \$P3D+ system has a lower percentage of 98% (CI = [96, 100]) - $t(8) = 0.59$, $p = 0.57$. This means that the \$P3D+ system is missing the best recognisable gun gesture and that is lowering its average. With the Right Handed L pose we can also determine that the recognition average is not significant between these systems while \$P3D+ has a significantly increased recognition rate when compared to \$P3D (92% CI = [82, 100]) and 78% CI = [72, 84]) respectfully - $t(8) = 1.00$, $p = 0.35$. For the Left Handed L pose the difference is extremely noticeable. For the \$P3D system, the average recognition rate is around 85% CI = [82, 88]). The \$P3D+ system achieves a higher recognition rate of 98% (CI = [95, 100]) with a significant difference between the 3D and 3D+ systems $t(6) = 5.97$, $p < 0.01$. This shows that when the user performs an L pose, the \$P3D+ system is going to recognise that pose easier.

Discussion

When differentiating \$P with \$P3D, the assumption was that \$P3D would outperform \$P. With the majority of gestures, \$P3D recognises the 3D point cloud with excellent gesture recognition average while \$P has high averages. The L Pose interestingly with the small database and the L and Rock Poses in the large database, the \$P gesture recognition system perform better than the 3D system.

When comparing \$P3D and \$P3D+, the improved gesture recognition system hypothetically yields slightly lower results based on the limitations of the training data gathered for testing. Our experiment shows that with some gestures, it actually increased the recognition rate by a small margin of 4.55%. The Left Hand L pose was the best result that achieved an increased recognition average of 13.17%. While this improvement increased the recognition average, we can claim that the confidence interval for the \$P3D+ gesture recognition average falls way out of the more condensed \$P3D versions interval as \$P3D factors in all gestures and gets the best-case scenario for each pose/gesture. However, the GUN gesture and the OK Pose did follow our hypothesis of having a smaller recognition rate compared to the traditional 3D gesture recognition. This is because of the small difference between left and right hand GUN gestures.

User Experiences

We demonstrated the system to several people during various events hosted by the Western Sydney University. While no formal questionnaires were given, we have documented the difficulties and differences between users who have used the system for the first time and some who have had experience with this device before. The users were a variety of ages ranging from primary school children, high school students, university undergraduates and some academic researchers. The users mostly inciated that they never used the LeapMotion system before.

From those who have no prior experience with the LeapMotion it was clearly that they found it difficult to first locate the LeapMotion system and use it properly. The main problem was the simplicity in the LeapMotions design causing users to not believe that the system could perform 3D hand recognition. Another issue was users who would immediately try to move their hand as close to the sensor as possible. While the LeapMotion can detect hands from a short distance, if the hands are too close to the IR cameras they could not distinguish the hands properly. Once instructed to move their hands upward, users understood the distance required for recognition and rarely brought their hands too close to the sensor again. Users were presented with either the default LotsOfBlocks demo or the Monsoon minigame developed for the first time users.

For the users who were presented with the Monsoon minigame, comments were made about what they were supposed to do while the minigame was playing. As the premise for the scene was to catch as many objects as possible, users found that the LeapMotion could not track their hands very well when they were cupped together. Some users decided to ignore grabbing the objects but rather tried to fling the objects around and as far as they could to see who could get the furthest.

The overall consensus with most individuals who used these demos was overwhelmingly positive. While some had seen the LeapMotion technology before or were not enthusiastic with the devices capabilities, the majority enjoyed using the tracking system.

6. CONCLUSIONS

In this paper, we have presented gesture recognition techniques that allowed recognition of both 2D and 3D gestures using point cloud gesture recognition. These techniques extrapolate gesture sets using

similar classifiers to recognise when to start looking for a gesture. We implemented tools that collected gesture data for recognition and processing. We also developed new 2D and 3D interactive environments that acted as a visual representation of the data.

We enhanced a 2D point cloud gesture recognition system into a 3D gesture recognition system that supports the data points from the LeapMotions hand, palm and fingers. We implemented four poses and two gestures to demonstrate the effectiveness of the new 3D gesture recognitions. We used a small database system for our gesture set in comparison to large database in existing systems.

Our experimental results showed that the 3D system outperformed the traditional 2D system in most cases as well as some small improvements on the $P3D+ in comparison to the original $P3D.

7. REFERENCES

[3Ge16] 3GearSystems. (Feb 2016). *Nimble VR*. http://nimblevr.com/

[Ant10] Anthony, L. and Wobbrock, J. O. A lightweight multistroke recognizer for user interface prototypes. In Proc. Graphics Interface 2010, Ottawa, Ontario, Canada, 2010.

[Ant12] Anthony, L. and Wobbrock, J. O. $N-protractor: a fast and accurate multistroke recognizer. In Proc. Graphics Interface 2012, Toronto, Ontario, Canada, 2012.

[Che14] Chen, K.-M. and Wong, S. K. Interactive Sand Art Drawing Using Kinect. In Proc. 7th International Symposium on Visual Information Communication & Interaction, pp. 78-87, 2014.

[Coo15] Cook, H., Nguyen, Q.V., Simoff, S., Trescak, T. and Preston, D. A Close-Range Gesture Interaction with Kinect. In Proc. IEEE International Symposium on Big Data Visual Analytics, Hobart, Australia, pp. 1-8, 2015.

[Ebe13] Ebert, L. C., Hatch, G., Thali, M. J., and Ross, S. Invisible touch—Control of a DICOM viewer with finger gestures using the Kinect depth camera. Journal of Forensic Radiology and Imaging, vol. 1, pp. 10-14, 2013.

[Kra11] Kratz, S. and Rohs, M. Protractor3D: A Closed-Form Solution to Rotation-Invariant 3D Gestures. *Intelligent user interfaces,* pp. 371, 2011.

[Kri04] Kristensson, P.-O. and Zhai, S. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. pp. 43, 2004.

[Lee13] Lee, U and Tanaka, J. Finger identification and hand gesture recognition techniques for natural user interface. *In Proc. Asia Pacific Conference on Computer Human Interaction,* pp. 274-279, 2013.

[Liu09] Liu, J., Zhong, L., Wickramasuriya, J. and Vasudevan, V. uWave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing, vol. 5, pp. 657-675, 2009.

[Pri16] PrimeSense and Apple. (Feb 2016). *OpenNI*. https://github.com/OpenNI/OpenNI

[Ope16] OpenKinect. (Feb 2015). *Libfreenect*. https://github.com/OpenKinect/libfreenect.

[Ren11] Ren, Z. Meng, J., Yuan, J. and Zhang, Z. Robust hand gesture recognition with kinect sensor. In Proc. ACM International Conference on Multimedia, pp. 759-760, 2011.

[Sha15] Sharp, T., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A. et al. Accurate, Robust, and Flexible Real-time Hand Tracking. *Computer Human Interaction,* pp. 3633-3642, 2015.

[Sho13] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A. et al. Real-time human pose recognition in parts from single depth images. *Communications of the ACM,* vol. 56, p. 116, 2013.

[Son13] Song, L., Hu, R., Xiao, Y., Gong, L. Real-Time 3D Hand Gesture Recognition from Depth Image. In Proc. the 2nd International Conference On Systems Engineering and Modeling (ICSEM-13), pp. 1134-1137, 2013.

[Tan11] Tang, M. Recognizing hand gestures with Microsoft's kinect. Department of Electrical Engineering, Stanford University, CA, USA, Technical Report, 2011.

[Vat12] Vatavu, R.-D., Anthony, L. and Wobbrock, J. O. Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes. In Proc. International Conference on Multimodal Interaction, p. 273, 2012.

[Wob07] Wobbrock, J.O., Wilson, A.D. and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. *User Interface Software & Technology,* pp. 159, 2007.

[Yin14] Yin, Y. Real-time continuous gesture recognition for natural multimodal interaction. PhD thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2014.

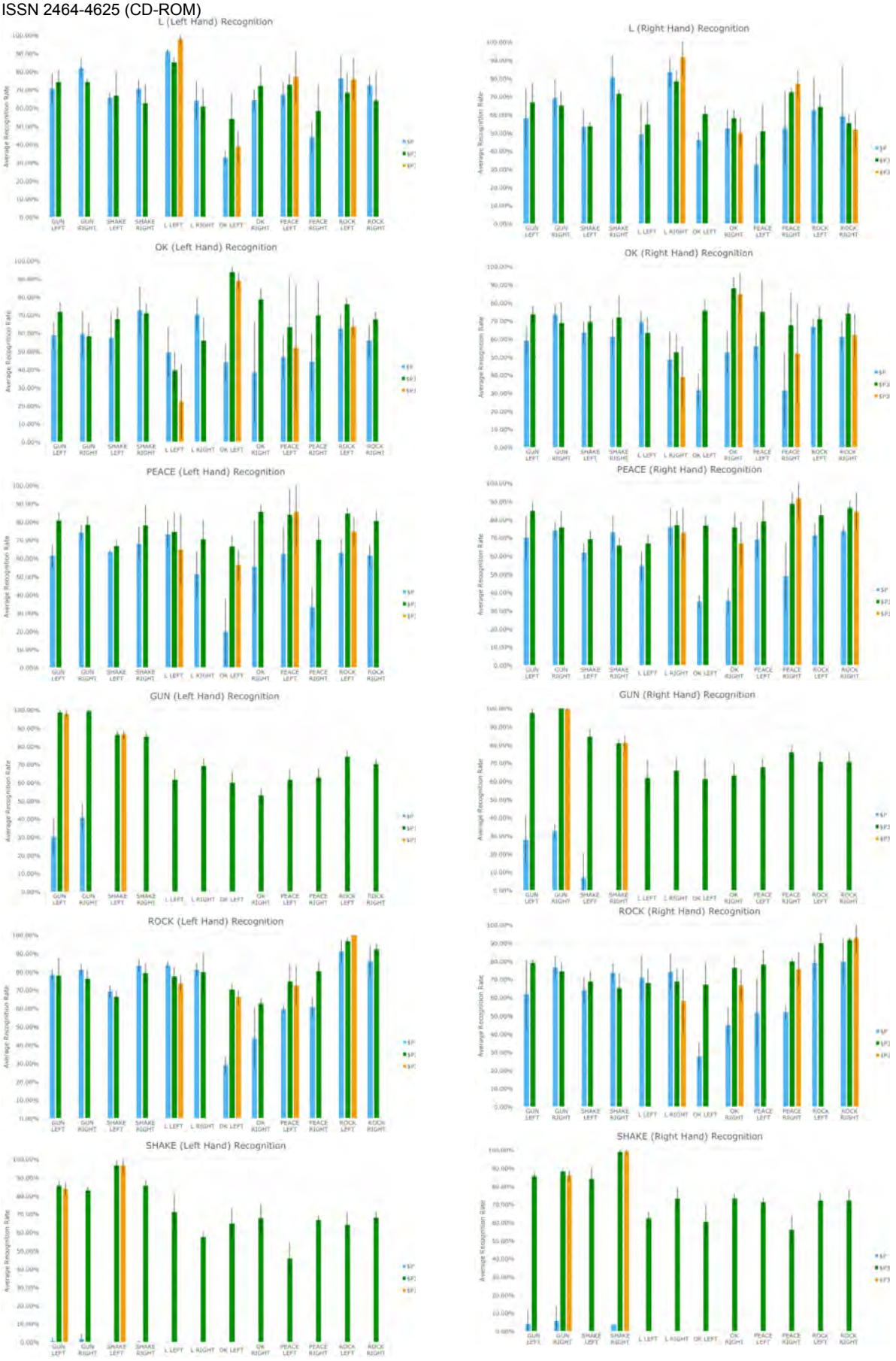


Figure 7. Gesture recognition averages on a large dataset with L, Ok, Gun, Peace, Rock and Shake on both left and right hands respectively.

Virtual Reality application to improve spatial ability of engineering students

Jordi Torner
Universitat Politècnica de Catalunya, BarcelonaTech
C/Urgell 187
08036, Barcelona, Spain
jordi.torner@upc.edu

Francesc Alpiste
Universitat Politècnica de Catalunya, BarcelonaTech
C/Urgell 187
08036, Barcelona, Spain
francesc.alpiste@upc.edu

Miguel Brigos
Universitat Politècnica de Catalunya, BarcelonaTech
C/Urgell 187
08036, Barcelona, Spain
miguel.brigos@upc.edu

ABSTRACT

We propose using Immersive Virtual Reality activities to improve the spatial ability of engineering students based on the study of solid geometry.

The work group is selected randomly from among all the students registered for the 1st term course Graphic Expression and Computer-Aided Design (GECAD) at the Barcelona College of Industrial Engineering (EUETIB).

A total of 60 participants completed three activities (6 h) in VR, using head-mounted display (HMD) glasses. Another group of students (30) made up the control group, which carried out only the learning activities that were common to all students, in a SolidWorks 3D non-immersive solid modeling software environment.

Spatial abilities are assessed using the Differential Aptitude Test: Spatial Relations Subset (DAT:SR) and the Purdue Spatial Visualization Test: Rotations (PSVT:R). Previous studies have demonstrated a close correlation between successful comprehension of the Graphic Engineering course contents and high scores on the DAT:SR test. The greatest correlation was found between the DAT:SR pre-test and the solid geometry exam (test and 3D modeling exercises).

We propose measuring spatial abilities before and after the classroom activities and looking for correlations between the spatial perception tests (DAT:SR and PSVT:R) and academic results in solid geometry. Furthermore, we also wish to determine the students' opinion with regard to the proposed activities.

This would permit us to recommend and incorporate the use of VR in order to improve spatial abilities, in particular for those students with lower levels of spatial abilities, as measured by a DAT:SR or PSVT:R.

Keywords

Spatial ability · Virtual reality (VR) · Teaching engineering · Head-mounted display (HMD).

1. INTRODUCTION

Our aim here is to apply Virtual Reality (VR) to learning processes in CAD, and to those contents related to space geometry.

According to [Hel92] VR uses hardware and software to create a sensation of immersion, navigation and manipulation.

VR is divided into three main categories: text-based, desktop and immersive VR.

Text-based networked VR is associated with environments in which communication takes place via texts. This type of VR has commonly been used in education and training [Pso95].

Desktop VR is an extension of interactive multimedia applications that incorporate three-dimensional images, but without being considered immersive.

Immersive VR is the application considered in this work. It consists of a combination of hardware, software and concepts that enable the user to interact with a computer-generated three-dimensional world [LA94].

There are different ways in which VR technology can aid learning.

VR is widely used in rehabilitation and surgical training [NGA+13].

Evidence exists [Bor16], [AAH+11], [LZXL05] that, when applied appropriately, VR can offer an effective means of improving certain skills. One example is the effective coordination of sensory motor skills using flight simulators for pilot training.

Some of the most interesting applications are those that enable students to visualize abstract concepts, such as chemical bonds [CTN15] or those that enable them to visit environments and interact with events or objects that would not otherwise be available due to restrictions imposed by distance, time or safety [Jav99].

[Win93] defines a conceptual framework for educational VR applications, and states that:

1. Immersive VR provides non-symbolic first-person experiences that are specifically designed to help students learn content.
2. These experiences cannot be obtained in any other way in formal education.
3. This type of experience constitutes a large part of our everyday interaction with the world around us, even though schools tend to promote symbolic third-person experiences.
4. Constructivism offers a theory upon which to develop educational VR applications.
5. The convergence of the theories of knowledge construction with VR technology permits promoting hands-on learning in virtual worlds, by making it possible to display information in a different way, and materialize abstract ideas that until now have been very difficult to represent.

[WHH+97] identifies the main contributions of VR as immersion, interaction or participation and motivation.

Pantelidis states that VR provides new ways and methods of visualization, is based on interaction and promotes active participation [PV10].

VR changes the way in which a student interacts with contents, as compared to traditional learning processes. Participants who interact with the virtual environment see immediate results, encouraging them to continue to interact with it, thus increasing their motivation.

[WM98] compared VR learning of an assembly procedure to other traditional media, such as paper and video. They obtained better long-term learning results with VR.

[SK03] compared the perception and understanding of spatial volumes using 2D, non-immersive VR (on a computer screen) and immersive 3D VR head-mounted environments. The students in an immersive VR environment attained a better understanding of volume and its components in three dimensions. Furthermore, the more complex the volume is, the better it is perceived using HMD.

In summary, there is a great deal of evidence to suggest that VR can be used successfully in learning processes. In many cases, the contribution of methodologies based on VR represents significant improvements in the comprehension of the course contents.

2. SPATIAL VISUALIZATION ABILITY

The ability to visualize space (or simply spatial ability or SA) is the capacity to understand and remember the spatial relationships between objects. More specifically, it is the ability to mentally maneuver two- and three-dimensional shapes [UMT+13].

Moreover, some authors [SS01], [MSB+11], [KJJ14] suggest the importance SA in the engineering design process and propose educational strategies to promote the development of this competence among students. The development of spatial capacity has always formed part of the Graphic Engineering curricula [MB05], [MG13]. One aim of introductory classes in graphic engineering is to increase the spatial capacity of students in order to construct cognitive representations of the geometric shapes and mentally maneuver them. To accomplish this, 2D and 3D representations are regularly used and students do exercises to obtain views, convert 2D into 3D, and vice versa. This makes it possible to obtain graphic representations of technical contents and to acquaint engineering students with the use of design resources. Increasing a student's spatial ability helps him/her better understand the concepts of solid geometry, which in turn, are used to design products, devices and systems.

SA is a complex human capacity that many researchers believe can be identified based on 2 or 3 components. Authors such as [PK82], [PH91] and [RG00], each cited by [MGK+13], identify three components of SA: spatial visualization, spatial orientation and spatial relation.

However, since spatial orientation and spatial visualization have factors in common, many researchers consider that there are really only two basic components of SA: spatial relation and spatial orientation [CG98], [HT11], [Moh08], [PRL+02], as referenced by [MGK+13].

Spatial relation is the capacity to mentally rotate an object around its axes. Spatial orientation is the ability to mentally maneuver or transform the representation of an object in another view.

In previous studies [TAB14], we have found a positive correlation between SA and academic performance in the study of solid geometry. The same is true for chemistry-related contents [MGK+13].

[Eli02] states that SA affects all daily activities, insofar as it refers to the ability to make mental representations and manipulate visual and spatial information. [LK83] divide SA into three subdomains: visualization, which includes complex tasks with numerous steps; spatial relations, which include simple tasks, such as quick mental rotation; and orientation, which includes tasks that involve imagining changes of perspective.

It would seem that the orientation and mental maneuvering abilities generally encompassed in the ability for spatial visualization rely on similar mental processes. [OSR+02] studied how people learn to

rotate objects. They compared the configurations of real and virtual objects (using HMD) and found a greater correlation with cognitive-analytic skills than with mental rotation abilities. They obtained the best results with participants who used HMDs, which led them to conclude that immersive VR is an excellent training tool for developing spatial capacity.

3. MEASURE OF SPATIAL ABILITY

From among the wide variety of tests used to measure the spatial visualization ability, we highlight four here that are frequently mentioned in the literature [DKSG06]:

- Differential Aptitude Test: Space Relations (DAT:SR); visualization, paper folding
- Mental Cutting Test (MCT); visualization, object cutting
- Mental Rotation Test (MRT); quick mental rotation
- Purdue Spatial Visualization Test: Rotations (PSVT:R)

The Differential Aptitude Test: Space Relations (DAT:SR) [BSW73] contains 50 items.

The task consists of selecting the correct representation of a 3D object from among four choices depicting the representation of a folded 2D shape. In one study [MGS98], it was found that student scores on the DAT: SR were the most important predictor of success in an engineering graphics course, as compared to three other spatial visualization tests.

The Mental Cutting Test (MCT) [Cee39] contains 25 items. Each question presents a shape that has been cut along a given plane. The objective is to select the resulting cut from among 5 alternatives.

The Mental Rotation Test (MRT), developed by [VK78], is used to assess a person's ability to visualize rotated solid objects. It consists of 20 items. Each question presents a shape with two correct and two incorrect choices. The objective is to identify which two alternatives are rotated images of the shape in question. A clear predecessor of the MRT is the Mental rotation of three dimensional objects by [SM71] used by [HLXB15].

The Purdue Spatial Visualization Test: Rotations (PSVT:R) was developed by [Gua77]. It contains 30 items. On this test, students are shown a reference object and then a view of the same object after it has been rotated in space. They are then shown a second object with a set of views and are asked which one matches the same rotation performed on the reference object [MY13].

[DKSG06] analyzed virtual reality (VR) and augmented reality (AR) as tools for use in SA training. In spite of non-conclusive results in terms of the improvement of SA with the tools used, improvements were shown on the four tests (used as pre- and post-tests) among the subjects.

The following tests were given:

- Differential Aptitude Test: Space Relations (DAT:SR); visualization, paper folding
- Mental Cutting Test (MCT); visualization, object cutting
- Mental Rotation Test (MRT); quick mental rotation
- Objective Perspective Test (OPT); orientation, change of perspective

In this study, [DKSG06] describe the limitation of not having a 3D test to measure SA, as traditional paper-based methods do not cover all the skills that come into play when working in a 3D environment.

In an earlier work [TAB14], we developed a model to assess SA in engineering students taking the first-year Graphic Expression and Computer-Aided design course.

In this work, an improvement in SA was observed following the use of 3D solid modeling software. Data from 812 first-year industrial engineering students in three colleges at Barcelona Tech (Universitat Politècnica de Catalunya) were analyzed.

The evolution of the results obtained on the Differential Aptitude Test: Spatial Relations Subset (DAT:SR) and Mental Rotation Test (MRT) were analyzed before and after the content material in computer aided design were studied.

We found the strongest correlation between the DAT pre-test and the solid geometry test. This led to the proposal of this article: to use VR to practice concepts of solid geometry, with the aim of improving academic results. DAT appears to be a good indicator of academic success, as it produced the greatest number of correlations.

On the other hand, MRT does not seem to be a good indicator, as it failed to provide any significant results. No strong correlations were found between the values of MRT and the solid geometry test.

Some data was collected to provide information about other related variables. Such as: age, sex, new/retaking student, engineering branch, route of entrance into the university, working while studying, previous years studying drawing and CAD software, sport practiced by the student, video game player, favourite video game type, internet user, right of left-handed and faculty of engineering.

Other relevant conclusions were:

Regarding the use of CAD software: better results were found for those students who had prior experience with this type of programs.

The data showed a slight improvement in the final DAT:SR score for those students who play sports as compared to those who do not.

Field of engineering: important differences were found among the specialties in which the students were enrolled. In particular, chemistry students obtained low scores.

This enabled us to design remedial educational activities for those students who required them.

4. HEAD-MOUNTED DISPLAY (HMD) APPLICATION

The device displays two almost identical photographs that differ by the point from which the photo is taken. When viewed by each of the eyes separately, the images simulate real vision and the brain composes a three-dimensional relief effect.

HMD devices currently reproduce two images with a slightly different focus on a monitor similar to that of a smartphone. The results cause a high level of immersion, while the movements of the head change the point of view of the receiver, enabling us to simulate movement in the scenario by means of a keyboard or joystick.

An attempt was originally made to use 3D modeling drawings from SolidWorks as the basis to create a version compatible with Oculus Rift, but the results obtained after converting the files were less than satisfactory.

The best resolution was obtained using Cinema4D in *.fbx format and importing it into the Unity game engine, in which the user movement and interaction commands had been programmed.

The scenarios corresponded to subsequent stages in the resolution of a solid geometry exercises. For learning purposes, the transitions between steps were animated following logical drawing processes.

Those animations were created using Unity's Legacy tool.

The symbols for geometric relationships and annotations were created in Photoshop (Figure 1).

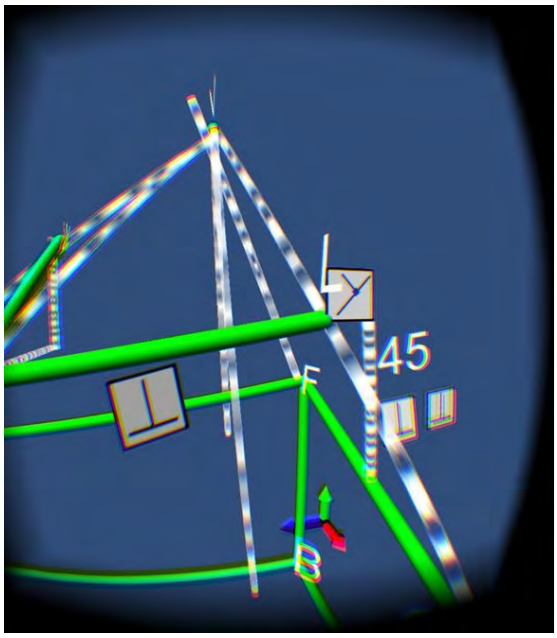


Figure 1. Screenshots from VR exercises.

5. METHODOLOGY

The investigation focused on the Graphic Expression and Computer-Aided Design (GECAD) course, specifically on the study of the SA developed and the assessment of the academic results in the solid geometry module.

The activities consisted of creating applications to model three-dimensional geometric shapes, introducing the concepts of geometry step by step. Students could interact freely with each scenario and move forward and backward through the sequence of steps.

Interaction takes place through the keyboard and visualization through an HMD headset.

The transitions between different states are created by means of provided animations that progressively build the elements by following the instructions for the exercise.

The concepts of solid geometry incorporated are synthesis, analysis and geometric metrics axioms, such as:

- In order for a straight line to be perpendicular to a plane, it needs only be perpendicular to two non-parallel straight lines found on the plane or parallel to it.
- If the angle between a straight line and a plane is θ, the angle between this straight line and the line perpendicular to the plane is complementary to θ: $(90-\theta)$.
- If a straight line has a slope of X%, this means that it has an angle with the horizontal plane of $\theta=\text{arctg}\ (X/100)$.

- If the angle between two planes is θ, the angle between two straight lines, each perpendicular to a plane, is the supplementary angle of θ: (180º-θ) (or θ, depending on where it is measured).
- The distance between two straight lines that cross one another is measured on the straight line perpendicular to both that intersects them.

The methodology can be summarized in the following steps:

1. Students in the experimental group and the control group take the Differential Aptitude Test: Space Relations (DAT:SR) and Purdue Spatial Visualization Test: Rotations (PSVT:R) prior to the activities. They also take the survey on controlled variables that can affect SA (1 h).
2. The students individually complete the exercises with the 3D modeling software SolidWorks (10 h). For example, drawing polyhedral shapes considering angles, distances and other geometrical relations between edges and faces.
3. The VR activities consist of the guided reading by the professor of the completed exercise. The professor addresses the concepts of solid geometry used in each step. The students have a few minutes to view the animation showing the construction of the geometric shape, and once the representation is finished, they can move freely throughout the scenario, using the keyboard options (6 h) (Figure 2).
4. Students in the experimental group and the control group take the Differential Aptitude Test: Space Relations (DAT:SR) and Purdue Spatial Visualization Test: Rotations (PSVT:R) after the VR activities. At the end, the groups that have worked in the VR also take the satisfaction survey (1 h). Control and experimental groups were formed randomly from all students enrolled in the course. We usually formed this division when new activities or methodologies are introduced in the course. In some cases, that is not possible because of the need to offer the same learning pursuits to all students without discrimination. In this case, both groups had similar results.
5. All the students are assessed on their knowledge of the solid geometry contents by means of a test and 3D modeling exercises similar to those done in class.
6. Finally, the analysis of the SA test data, the controlled variable surveys and satisfaction surveys and the academic results obtained in the solid geometry module enable us to examine the correlations and the strongest determining factors in order to obtain good academic results and propose VR activities to improve the levels of SA obtained on the tests.

Figure 2. VR sessions.

6. RESULTS

The results of this study are exploratory in nature by the small size of the sample and cannot be generalized to the population as a whole. Because of that, we do not discus here values when comparing experimental and control groups' results. The same can be said about the data collected from other variables.

We urge further work analyzing the influence of other factors on the increase in SA, such as: gender, prior experience with this type of programs, play sports or faculty of engineering.

With this in mind, we would like to highlight the following findings in the experimental group:

Those students with less spatial ability as measured by a PSVT:R (Table 1) or DAT:SR pre-test (Table 2) are those who show the greatest degree of improvement on the post-tests after having completed the VR activities.

pre-test PSVT (Results)	Degree of improvement (Average)
1st set (0-15)	14,3
2nd set (16-20)	2,2
3rd set (>21)	1,02

Table 1. PSVT results vs degree of improvement.

pre-test DAT (Results)	Degree of improvement (Average)
1st set (0-30)	12,4
2nd set (31-50)	4,9
3rd set (>50)	0,6

Table 2. DAT results vs degree of improvement.

The greatest correlation (R=0.323) was found between the PSVT:R pre-test and the solid geometry exam (Test and 3D modeling exercises (Table 3).

These results corroborate those obtained by [SB00] at Michigan Technological University (MTU) with regard to the interest in using PSVT:R as a predictor of academic results.

Model	R	R Square	Adjusted R	Std. Error
1	,323	,104	,085	2,35307

Table 3. Correlation PSVT:R pre-test vs. solid geometry exam

The experimental group and the control group showed no significant differences.

The evaluation by the students of the incorporation of VR/HMD activities is positive (Table 4). They have the perception that the system is easy to use, enables them to better understand the contents presented and they consider it to be useful. Therefore, in agreement with the findings of [LL12], the use of VR motivates students during their learning process.

The immersion system is easy to use
Strongly disagree 0%
Disagree 0%
Neither agree nor disagree 26%
Agree 63%
Strongly agree 11%
The content provided is easy to understand
Strongly disagree 0%
Disagree 11%
Neither agree nor disagree 58%
Agree 32%
Strongly agree 11%
Immersion system provides useful content
Strongly disagree 0%
Disagree 0%
Neither agree nor disagree 16%
Agree 47%
Strongly agree 37%

Table 4. Satisfaction survey results

7. LIMITATIONS & FUTURE WORKS

Some results obtained from the data analysis are indecisive in terms of the correlation between the VR activities carried out and the improvement in spatial abilities. The activity also failed to have an impact on academic results.

In our opinion, this may be caused by several factors:

1. The small size of the experimental (60) and control (30) groups.
2. The short duration of the VR/HMD activity, which consisted of only two sessions of three hours each. In previous studies, we obtained important increases (8 points) in scores on the Differential Aptitude Test: Spatial Relations Subset (DAT:SR) after holding numerous 3D solid modeling sessions (15 h).
3. The impact of other variables on the learning process, the definition and influence of which should be analyzed in future works.

In any case, the results obtained confirm the interest in using VR to develop SA in engineering students. In future studies, we intend to expand the size of the sample and the diversity of the populations analyzed, in order to make the results more generalizable. In this regard, work is underway to incorporate conventional mobile devices, which when fastened to a simple support structure similar to a pair of glasses and connected to a control device (numeric keypad or joystick) would make it possible to extend the experience to a large number of students at several colleges.

The basic assertion that we would like to test in future work is that students who use the VR show a significant improvement in spatial abilities as compared to those who do not use it.

Another concept related to VR is augmented reality (AR), which uses a combination of the user's physical environment and real-time interactive computer representations [ADD+01], [Bon01], [WTA11].

VR immerses the user in a digital 3D environment where he/she cannot see the real world. Just the opposite, AR enables the user to see the real world, with some virtual objects superimposed on it. AR thus complements reality instead of completely replacing it. Virtual and real objects are perceived in the same space. AR is currently being applied in education and training with good results [GGW+13], [Lee12], [WLC13], [AAP+12].

Both VR and AR environments could be applied to our study, although ultimately the decision was made to use an immersive VR environment due to the abstract nature of the solid geometry course content being taught. One proposal for future work could be to incorporate geometry topics that combine both real and synthetic images. This could be extremely useful for design validation in engineering design courses.

Other additions for future works could be:

The use of Mental Cutting Test (MCT) and the study of its effectiveness in order to improve the model as a predictor of learning outcomes.

The use of 3D tests. Some cases already exist in which immersive 3D tests have been applied, such as the

Virtual Reality Spatial Rotation (VRSR) system [RBN+98], which make it possible to administer the Mental Rotation Test (MRT) [SM71] directly in 3D, providing improvements over the administration of the paper-and-pencil MRT following VRSR training.

Geometric immersive 3D graphic design software could also be added. [DKSG06] use drawing software that makes it possible to generate 3D models, move around them and modify them in real time. Visualization relies on HMD glasses.

Within the plans for continuing this study is the incorporation of some of these facilities in the near future.

8. CONCLUSIONS

A pilot study was developed using immersive VR to study solid geometry in a 1st term course of engineering. The activities proposed for all students were 3D modeling exercises in Solidworks, and the experimental group completed three activities (6 h) in VR, HMD glasses.

A methodology was applied to assess SA in engineering students previously developed by our research group at Barcelona Tech (Universitat Politècnica de Catalunya). We have incorporated a new evaluation tool in this study: Purdue Spatial Visualization Test: Rotations (PSVT:R). This test appears to be a good indicator of academic success.

Spatial abilities are assessed before and after the classroom activities using the Differential Aptitude Test: Spatial Relations Subset (DAT:SR) and the Purdue Spatial Visualization Test: Rotations (PSVT:R). Despite the fact that all students have improved their results, the greatest degree of improvement on the post-tests was achieved by the students with less initial spatial ability.

The greatest correlation was found between the PSVT:R pre-test and the solid geometry exam (test and 3D modeling exercises). These results corroborate previous studies using PSVT:R as a predictor of academic results.

The evaluation by the students of the incorporation of VR/HMD activities is positive. After the experience, we strong recommend the use of VR in order to improve spatial abilities of engineering, particularly those students showing difficulties in spatial visualization.

9. REFERENCES

[AAH+11] Amin Nur Yunus, F.; Abd Baser, J.; Hadi Masran, S.; Razali, N.; and Rahi, B. (2011). Virtual Reality Simulator Developed Welding Technology Skills. Journal of Modern Education Review, ISSN 2155-7993, USA, Volume 1, No. 1, pp. 57-62

[AAP+12] Aziz, N. A. A.; Aziz, K. A.; Paul, A.; Yusof, A. M.; and Noor, N. S. M. (2012, February). Providing augmented reality based education for students with attention deficit hyperactive disorder via cloud computing: Its advantages. In Advanced Communication Technology (ICACT), 2012. 14th International Conference on Advanced Comunication Technology (pp. 577-581). IEEE

[ADD+01] Azuma, R.; Baillot, Y. R.; Behringer; Feiner, S.; Julier, S.; and MacIntyre, B. (2001). Recent advances in augmented reality. Computer Graphics and Applications, IEEE, vol. 21, pp. 34-47

[Bon01] Bonsor, K. (2001). How Augmented Reality Will Work. http://howstuffworks.com/augmented-reality.htm

[Bor16] Borsci, S. et al (2016). Effectiveness of a multidevice 3D virtual environment application to train car service maintenance procedures. Doi: 10.1007/s10055-015-0281-5

[BSW73] Bennett, G. K.; Seashore, H. G.; and Wesman, A. G. Differential Aptitude Tests, Forms S and T. The Psychological Corporation, New York, 1973

[Cee39] CEEB College Entrance Examination Board. Special. Aptitude Test in Spatial Relations MCT. CEEB, 1939

[CG98] Coleman, S. and Gotch, A. (1998). Spatial perception skills of chemistry students. Journal of Chemical Education, 75, 206-209

[CTN15] Carlisle, D.; Tyson, J. F.; and Nieswandt, M. (2015). Fostering spatial skill acquisition by general chemistry students. Chemistry Education Research and Practice

[DKSG06] Dünser, A.; Kaufmann, H.; Steinbügl, K.; and Glück, J. (2006). Virtual and Augmented Reality as Spatial Ability Training Tools. 7th ACM SIGCHI New Zealand chapter's international conference on Computer-Human Interaction. Pages 125-132

[Eli02] Eliot, J. (2002). About spatial intelligence: I. Perceptual and Motor Skills, 94, 2, 479-486

[GGW+13] Gavish, N.; Gutiérrez, T.; Webel, S.; Rodríguez, J.; Peveri, M.; Bockholt, U.; and Tecchia, F. (2013). Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. Interactive Learning Environments (ahead-of-print), 1-21. DOI: 10.1080/10494820.2013.815221

[Gua77] Guay, R. B. (1977). Purdue spatial visualization test: rotations. West Lafayette, IN. Purdue Research Foundation

[Hel92] Helsel, S. (1992). Virtual Reality and Education, Educational Technology, 32(5), pp. 38-42

[HLXB15] Hawes, Z.; LeFevre, J. A.; Xu, C.; and Bruce, C. D. (2015). Mental Rotation With Tangible Three-Dimensional Objects: A New Measure Sensitive to Developmental Differences in 4-to 8-Year-Old Children. Mind, Brain, and Education, 9(1), 10-18

[HT11] Harle, M. and Towns, M. (2011). A review of spatial ability literature, its connection to chemistry, and implications for instruction. Journal of Chemical Education, 88, 351-360. DOI: 10.1021/ed900003n

[Jav99] Javidi, G. (1999). Virtual reality and education, Master Thesis, University of South Florida

[KJJ14] Katsioloudis, P.; Jovanovic, V.; and Jones, M. (2014). A Comparative Analysis of Spatial Visualization Ability and Drafting Models for Industrial and Technology Education Students

[LA94] Loeffler, C. E. and Anderson, T. (eds.) (1994). The Virtual Reality Casebook, New York:Van Nostrand Reinhold

[Lee12] Lee, K. (2012). Augmented reality in education and training. TechTrends, 56(2), 13-21

[LK83] Lohman, D. F. and Kyllonen, P. C. (1983). Individual differences in solution strategy on spatial tasks. Individual differences in cognition. Dillon, D. F. and Schmeck, R. R. (eds.). Academic Press: New York, 105-135

[LL12] Lau, K. W. and Lee, P. Y. (2012). The use of virtual reality for creating unusual environmental stimulation to motivate students to explore creative ideas. Interactive Learning Environments (ahead-of-print), 1-16. DOI: 10.1080/10494820.2012.745426

[LZXL05] Li, L., Zhang, M., Xu, F., Liu, S.(2005). Ert-vr: an immersive virtual reality system for emergency rescue training. Virtual Real. 8(3), 194–197

[MB05] Miller, C. L. and Bertoline, G. R. (2005). Spatial Abilities and Virtual Technologies: Examining the Computer Graphics Learning Environment, iv, pp. 992-997, Ninth International Conference on Information Visualisation (IV '05)

[MG13] Marunic, G. and Glazar, V. (2013). Spatial ability through engineering graphics education. International Journal of Technology and Design Education, 23(3), 703-715

[MGK+13] Merchant, Z.; Goetz, E. T.; Keeney-Kennicutt, W.; Cifuentes, L.; Kwok, O.; and Davis, T. J. (2013). Exploring 3-D virtual reality technology for spatial ability and chemistry achievement. Journal of Computer Assisted Learning, 29, 579-590, John Wiley & Sons Ltd

[MGS98] Medina, A. C.; Gerson, H. B. P.; and Sorby, S. A. (1998). Identifying gender differences in the 3-D visualization skills of engineering students in Brazil and in the United States. Proceedings of the International Conference for Engineering Education 1998, Rio de Janeiro, Brazil

[Moh08] Mohler, J. L. (2008). A review of spatial ability research. Engineering Design Graphics Journal, 72, 19-30

[MSB+11] Metz, S. S.; Sorby, S. A.; Berry, T. S.; Seepersad, C. C.; Dison, A. M.; Allam, Y. S.; and Leach, J. A. (2011). Implementing ENGAGE Strategies to Improve Retention: Focus on Spatial Skills Engineering Schools Discuss Successes and Challenges. American Society for Engineering Education

[MY13] Maeda, Y., & Yoon, S. Y. (2013). A meta-analysis on gender differences in mental rotation ability measured by the Purdue spatial visualization tests: Visualization of rotations (PSVT: R). Educational Psychology Review, 25(1), 69-94.

[NGA+13] Nagendran, M., Gurusamy, K. S., Aggarwal, R., Loizidou, M., & Davidson, B. R. (2013). Virtual reality training for surgical trainees in laparoscopic surgery. Cochrane Database Syst Rev, 8.

[OSR+02] Oman, C.; Shebilske, W. L.; Richards, J. T.; Tubré, T. C.; Beall, A. C.; and Natapoff, A (2002). Three Dimensional Spatial Memory and Learning in Real and Virtual Environments. Spatial Cognition and Computation, 2, 355-372

[PH91] Pellegrino, J. and Hunt, E. (1991). Cognitive models for understanding and assessing spatial abilities. In Rowe, H. (ed.), Intelligence: Reconceptualisation and measurement (pp. 203-225). Melbourne: ACER and Hillsdale, NJ: Lawrence Erlbaum Associates

[PK82] Pellegrino, J. W. and Kail, R. (1982). Process analyses of spatial aptitude. In Sternberg, R. (ed.), Advances in the psychology of human intelligence (vol. 1, pp. 311-366). Hillsdale, NJ: Erlbaum

[PRL+02] Piburn, M.; Reynolds, S.; Leedy, D.; McAuliffe, C.; Birk, J.; and Johnson, J. (2002). The hidden earth: Visualization of geologic features and their subsurface geometry. Paper presented at the National Association for Research in Science Teaching (NARST) annual meetings, April 7-10,2002, New Orleans, Louisiana

[Pso95] Psotka, J. (1995). Immersive Tutoring Systems: Virtual Reality and Education and

Training. Instructional Science 23, pp. 405-431, 1995

[PV10] Pantelidis, Veronica S. (2010). Reasons to Use Virtual Reality in Educationand Training Courses and a Model to Determine When to Use Virtual Reality. Themes in Science and Technology Education. Special Issue, pages 59-70. Klidarithmos Computer Books

[RBN+98] Rizzo, A. A.; Buckwalter, J. G.; Neumann, U.; Kesselman, C.; Thiebaux, M.; Larson, P.; and Van Rooyen, A. (1998). The Virtual Reality Mental Rotation Spatial Skills Project. CyberPsychology and Behavior, 1, 2, 113-120

[RG00] Robichaux, R. R. and Guarino, A. J. (2000). Predictors of visualization: A structural equation model. Annual Meeting of the Mid-South Educational Research Association, Bowling Green, KY

[SB00] Sorby, S. A. and Baartmans, B. J. (2000). The development and assessment of a course for enhancing the 3-D spatial visualization skills of first year engineering students. Journal of Engineering Education, 89(3), 301-307

[SK03] Schnabel, M. A. and Kvan, T. (2003). Spatial understanding in immersive virtual environments. International Journal of Architectural Computing, 1, 4, 435-448

[SM71] Shepard, R. N. and Metzler, J. (1971). Mental rotation of three dimensional objects. Science, 171, 972, 701-3

[SS01] Strong, S. and Smith, R. (2001). Spatial visualization: fundamentals and trends in engineering graphics. Journal of Industrial Technology, vol. 18, No. 1

[TAB14] Torner, J.; Alpiste, F.; Brigos, M. (2014) Spatial ability in computer-aided design courses. Computer-aided design and Applications. Vol. 12, num. 1, p. 1-9 DOI: 10.1080/16864360.2014.949572

[UMT+13] Uttal, D. H.; Meadow, N. G.; Tipton, E.; Hand, L. L.; Alden, A. R.; Warren, C.; and Newcombe, N. S. (2013). The malleability of spatial skills: a meta-analysis of training studies. Psychological bulletin, 139(2), 352

[VK78] Vanderberg, S. G. and Kuse, A. R. (1978). Mental rotations, a group test of three-dimensional spatial visualization. Perceptual and motor skills. Volume 47, issue , pp. 599-604. DOI: 10.2466/pms.1978.47.2.599

[WHH+97] Winn, W.; Hoffman, H.; Hollander, A.; Osberg, K.; Rose, H.; and Char, P. (1997). The Effect of Student Construction of Virtual Environments on the Performance of High- and Low-Ability Students. Human Interface Technology Laboratory. http://www.hitl.washington.edu/publications/r-97-6/

[Win93] Winn, W. (1993). A conceptual basis for educational applications of virtual reality. HITL Laboratory. http://www.hitl.washington.edu/publications/r-93-9

[WLC13] Wu, H. K.; Lee, S. W. Y.; Chang, H. Y.; and Liang, J. C. (2013). Current status, opportunities and challenges of augmented reality in education. Computers & Education, 62, 41-49

[WM98] Waller, D. and Miller, J. A. Desktop virtual environment trainer provides superior retention of a spatial assembly skill. Poster presented at ACM SIGCHI ’98, Los Angeles, CA

[WTA11] Wither, J.; Tsai, Y. T.; and Azuma, R. (2011). Indirect augmented reality. Computers & Graphics, 35(4), 810-822

Real-Time Rendering of Continuous Levels of Detail for Sparse Voxel Octrees

Szymon Jabłoński
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
s.jablonski@ii.pw.edu.pl

Tomasz Martyn
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
martyn@ii.pw.edu.pl

ABSTRACT

In this paper, we present a novel approach to real-time, continuous and symmetrical level of detail (LOD) management of a 3D object represented by a sparse voxel octree (SVO). We propose a new method for continuous and symmetrical transition between two detail levels. The method is based on a SVO representation extended by redundant, helper nodes which are used to achieve a proper interpolation of geometry and material data. We extend redundant nodes with a transition direction attribute. Additional memory requirements are minimized by storing indices in a direction vector lookup table in object space. The new method is applied for an accurate evaluation of the required LOD. It uses an image-based evaluation function, i.e. the standard level transition function based on camera distance is extended by the real-time calculation of the current LOD pixel fill rate. We extend typical level transition function based on distance with real-time calculation of the current LOD pixel fill rate. Two different image based methods of SVO node pixel fill rate calculations using compute shaders or GPU queries and parallel reduce are presented. The developed LOD management algorithm is applicable for a raytracing and a rasterization-based rendering pipeline. The LOD transition algorithm allows to perform a dynamic and continues control of the SVO based objects which have not been available in other works. Moreover, the proposed fading algorithm based on the fade out direction and scaling allows for a LOD change without any graphical artifacts or loss of the virtual scene immersion.

Keywords

Computer graphics, level of detail, sparse voxel octree, voxel rendering, parallel reduce, image processing

1 INTRODUCTION

Computer graphic engines are perfect examples of the soft real-time systems [Tanen07]. A key requirement for the real-time system is the processing time measured in tenths of seconds or shorter. Interactive graphic applications, such as computer games or virtual reality, require that all necessary logic computation and rendering is performed within a few milliseconds. Due to the limited memory of GPUs, achieving satisfactory rendering results requires an implementation of several optimization methods in our graphics engine pipeline.

An important observation is that with the perspective projection objects that are far away from observer appear on the screen much smaller than objects that are near the observer. This implies that they can be rendered with less geometrical or material details. Thus, different techniques for controlling the object's current level of detail (LOD) have been developed to adapt the object's complexity to their importance within the virtual scene.

The LOD is one of the oldest problems in computer graphics. It was first presented in an article by James H. Clark in 1976 [Clark90] and was defined as the complexity of the 3D object located at a suitable distance from the observer. The main aim of using LOD control algorithms is to increase rendering efficiency by minimizing the data consumption, e.g. a number of polygons or voxels. Increasing computation power of today's GPUs allowed game developers to create highly detailed virtual scenes represented by millions of polygons. As a consequence, efficient LOD management algorithms are needed more than ever.

The tendency in recent years has been to improve the features of continuous LOD algorithms by using the possibilities offered by the graphics hardware, such as the tessellation shaders [Schaf14]. The main problem with these methods is that there are no special-

ized shaders to decimate geometry on GPU. Moreover, there is still the need to store a complex geometry in the VRAM. As a result, the most commonly used solution in today's graphics engines is to prepare several objects with different LODs and change them in real-time based on their importance within the virtual scene [Lueb02].

In this paper, we present a novel approach to real-time, continuous and symmetrical management of a 3D object LOD represented by the sparse voxel octree (SVO).

2 RELATED WORK

There is a wide selection of literature on performing the LOD evaluation and control algorithms. Over the years, many methods of geometry simplification and continuous LOD controlling have been developed. However, most of them are actually using polygonal representation of geometry and are connected to the LOD of virtual scene terrains based on height fields. We will focus on the papers that are most directly related to our work.

As we mentioned in the introduction, the most commonly used method is based on preparing a finite count of objects in different LODs by artists and change the currently used one based on the specified distance function [Lueb02]. It is very easy to implement and control in real-time. However, the objects representing a different LOD have to be stored in GPU memory. Alternatively, a streaming functionality must be implemented in the graphic engine. Moreover, the continuous transition between triangle based objects with a significantly different number of polygons is quite difficult.

Schoeder et al. introduced algorithms to decimate a triangle mesh [Schroeder92]. Rossignac and Borrel extended their approach by developing vertex clustering method that uses the bounding box of the source mesh divided into the grid with all of the vertices in a given cell replaced with a single vertex [Rossignac93]. Other groups of algorithms use an iterative approach based on primitive simplification operators [Lueb02] such as edge collapse or vertex removal. One of the commonly applied iterative decimation technique is the QSlim algorithm [Garland97]. In this method the pair collapse operator is iteratively replacing two vertices with one, causing neighboring faces to become degenerated. However, the mesh simplification has been traditionally viewed as a non-interactive process not readily amenable to the GPU acceleration. Using the modern programmable GPU, many algorithms have been extended and developed on the GPU. Ramos et al. developed a method of continuous geometry complexity controlling based on GPU triangle strip operations [Ramos06]. The vertex clustering method has been extended by using geometry shaders and a general-purpose data structure called the probabilistic octree. It enabled a simultaneous construction of multiple LODs and out-of-core simplification of extremely large polygonal meshes [DeCoro07, Schiffner15, Willmott11].

Modern GPUs offer functionality to increase geometry complexity in real-time [Schaf14]. By using programmable tessellation shaders, we can increase the number of mesh triangles by dividing triangle patches. It is possible to achieve full control of how the object geometry is dived and to calculate all required data attributes such as normal vector and texture coordinates. This method has been effectively used in case of terrain rendering based on height field [Ripol12]. Even by using low-resolution height map one can create highly detailed terrain with continuous, distance dependent LOD. Using tessellation shaders one can increase the complexity of the processing object exclusively on the selected areas. Unfortunately, today's graphics adapters do not offer any programmable shaders to decrease the complexity of processing object. In order to create a symmetrical method to increase and decrease object LOD, it is necessary to combine simplification and tessellation approach into one algorithm.

Although all of the presented methods propose interesting ideas related to the LOD management, none of them is able to provide a symmetrical, continuous and universal LOD control. A common feature of all of the presented methods is the polygonal representation of 3D objects. The polygonal representation does not provide any hierarchical information. A solution to that problem is the usage of the voxel representation. The SVO is the current standard method for representing and rendering voxels [Laine10, Bau11, Wil13]. Cyril Crassin was able to perform visualization of global illumination based on an SVO and voxel cone tracing [Crassin11]. The SVO is a hierarchical structure that, in addition to the significant voxel memory compression, offers object's LODs. Based on the voxelization resolution one can calculate the maximum tree depth which is the number of the object's LODs. A dynamic LOD is quite often mentioned in various articles. However, no one has ever described an algorithm showing how to change the current LOD and how to perform a transition between the levels in a continuous way.

3 LOD MANAGEMENT ALGORITHM

This section describes the fundamental features of LOD management algorithms. All algorithms can be divided into the two main components:

- **LOD evaluation** — in this part one needs to determine when to change the object's current LOD. The initiation of the change and its direction depends on, for example, the distance between the object and the observer or object size on the screen.

- **LOD transition** — this part of the algorithm deals with the way of how the current LOD is changing. In the discrete model algorithm, the change is realized by simple swapping of the object geometry or material data. In the case of the continuous algorithm in order to achieve proper object transition, one needs to implement the interpolation between two selected LODs.

The features of the developed algorithm are as follows:

- The 3D object is represented by an SVO.
- The LOD management algorithm is independent of the rendering method. It can be used with both the ray tracing approach and the voxel visualization achieved with the triangle rasterization pipeline.
- Helper data needed for the algorithm execution are minimized. In the case of ray tracing visualization, all necessary data can be calculated on the fly.
- The LOD transition is done smoothly using interpolation between two levels of SVO.
- The object geometry and material data LODs are increasing or decreasing symmetrically.

4 LOD EVALUATION

The most commonly used measurement to evaluate the object's current LOD is the distance between the object and the observer position. However, to do that, the scene designer must manually find the proper distance for each object to achieve satisfying results. The algorithm proposed in this paper estimates the distance using a newly developed image-based method utilizing the object rendering results achieved in pre-processing stage or by using rendering results of the previous frame.

4.1 Continuous evaluation function

In general, the LOD evaluation function can be expressed as:

$$y = f(x) \tag{1}$$

where:

y = object LOD
x = evaluation parameter

Based on the computed distance between the object and observer position the current LOD is determined in a way presented in Fig. 1. For simplicity, the linear dependence between the distance value and the LOD is assumed.

In this case Eq. 1 can be written in the following form:

$$y = max\left(0, min(N - \frac{dist}{x}, N)\right) \tag{2}$$

where:

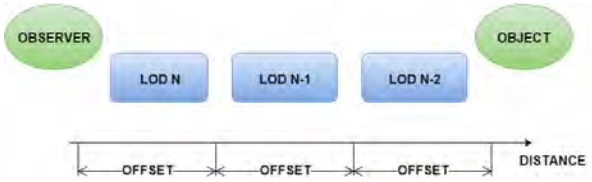


Figure 1: Linearly changing object LOD based on the distance.

y = object LOD
N = number of object LODs
$dist$ = distance between object and observer
x = defined distance offset between adjacent LODs

The computation results must be clamped to <*0,N*> in order to operate only on the existing object LODs. Assuming $N = 4$, $x = 2.0$ and $dist = 2.5$ we get:

$$y = max\left(0, min(4 - \frac{2.5}{2.0}, 4)\right) = 2.75 \tag{3}$$

Using mathematical round or truncate for the obtained result, we gain an integer value which represents the discrete LOD index. In addition, the floating result gives the possibility to control the continuous transition weight between two LODs.

The evaluation function presented above is based on the linear dependence between the distance value and the LOD. However, this approach can hardly give satisfying results on the virtual scene viewed with the perspective projection. More accurate results could be achieved using exponential or power functions.

The biggest problem with distance based evaluation functions is that there is no exact relationship between the distance and the resulting image. The rendering result will be different, but there are no algorithmic tools to describe scene complexity changes. In the next section, we propose a LOD evaluation method considering the image pixels as the main information carrier [Shannon48].

4.2 Pixel fill rate based evaluation function

Using compute shaders, data obtained from the rendering pass or GPU query objects according to the visualization algorithm, we can calculate how many pixels are filled by the draw operation [Wright10]. Thus, we can calculate how many pixels will be filled by some part of the object. The question is when we actually want to change the current LOD. If the object is so far away from the observer that there is no need to render it with the current LOD because we won't see any differences in rendering results, we can optimize the rendering process by decreasing the current LOD. Similarly, when the object is close enough to be rendered with a more complex geometry or material details we increase the current LOD.

5 PATTERN NODE SELECTION METHOD

In order to decide if the object is rendered with enough details, we analyze the rendering results of the smallest part of the object, the SVO node nearest to the observer position. In this paper, we define this special node as a *pattern node*. In order to find the pattern node, we propose two methods which differ in calculation precision and computing performance.

5.1 Approximate pattern node selection

The first method is based on a simple and naive approximation. For simplicity, we assume that the pattern node is exactly in the object bounding volume center position (even if actually there is no node in the selected 3D space). We can consider this method as an extension of the distance based evaluation method.

Having the object bounding volume and the node size of the current LOD, we render a single node off-screen. If our rendering pipeline is based on the ray tracing approach, we simply render the root node of the object scaled to the size of the current LOD node. In order to optimize the additional ray tracing step, we can perform ray tracing to a smaller render target than screen size. The minimum size of the target viewport can be easily calculated based on the object bounding volume and the camera matrices. Using, for example, atomic counters, we can calculate how many pixels will be filled by the draw operation. In the case of using a polygonal representation of the object, the same results can be achieved using GPU query objects.

As we mentioned before, it is a naive approach but can give satisfying results, especially on a low-power target like mobile devices.

5.2 Accurate pattern node selection

The accurate approach is much more advanced than the previous one. We want to find which SVO node of the object is the biggest on the screen. In other words, which one fills the most pixels of the resulting image. We must find the node in the current level of the SVO that is the closest to the observer position. We can calculate it on CPU by performing a simple software renderer but it is a computationally expensive solution hardly executable in real-time.

In order to obtain an accurate result, we need to perform the additional rendering and computation pass before the SVO visualization step. The accurate selection method will depend on the rendering algorithm. In this section we describe the pattern node selection for the ray tracing approach as well as for the triangle-based rasterization pipeline.

The foundation of the accurate pattern node selection algorithm is finding an SVO node with the smallest depth value for the current viewpoint. It means that before performing the final rendering, we need to find the node and count have many pixels will be filled by the node. The proposed solution can be implemented in a various way. In our work, we decided to use compute shaders. However, the developed solution can be implemented in other GPGPU interfaces such as CUDA or OpenCL.

5.2.1 Depth and node id texture generation

We propose an image based solution to find the pattern node utilizing the node depth information. By using the depth buffer, we find which pixel of the resulting image belongs to the object closest to the observer. The first step of our algorithm is an off-screen z-pass rendering. We use single channel render target texture with e.g. R32F or R16F internal format and save the linearized depth information. Further, the depth information has to be correlated to the SVO nodes.

The first step of our method requires defining unique ids for all SVO nodes. To optimize the node finding operation with a specified index it is recommended to create a lookup table for the tree nodes. Then, we extend the first pass of the algorithm by the saving node indices to the second texture channel. With the depth-only solution we could use floating point textures but unfortunately, we cannot save and retrieve integer or unsigned integer data from a float texture without data loss. A high-resolution voxel object will have ids counted in millions and we could lose the information. To solve this problem we use the unsigned integer type texture. Moreover, when saving the linearized depth information we perform normalization to the defined data range by multiplying depth values. We use a texture in the RG32UI internal format.

5.2.2 Minimum depth node seeking

The next step of our algorithm is to find the minimum depth from the rendered texture. If we need this information on CPU, we could transfer the texture data from GPU memory into the RAM. Due to the high cost of this data transfer, we might not be able to evaluate the LOD in real-time. In order to achieve real-time results, we can use two different solutions.

First of them is based on the parallel reduce algorithm on GPU [Buck04]. Using a compute shader, we create a new texture with half size of the source texture. Then, using a simple code we seek for the smallest depth value of N neighbors and store it in the new texture. By the defined number of iterations, we create a small texture with candidate nodes. We perform the parallel reduce algorithm until we create a 1x1 resolution texture and transfer it or read it on CPU. It is also possible to stop the iteration when our texture is small enough for being transferred to the CPU efficiently. In our implementation, we used the 1280x720 render target and 4

iterations of the parallel reduce algorithm resulting in a 80x45 resolution texture. We iterate through and find the pattern node on the CPU.

With the parallel reduce algorithm we can efficiently find the pattern node but we recommend an alternative solution. When performing the rendering operation, we can save the minimum depth of the rendered nodes with the corresponding node id by using the atomic operations and shader storage buffer objects. Thanks to that, we can use this information in the next step of our algorithm. Additionally, the access to the data stored in the shader storage buffer object on CPU side is very efficient. With a compute shader, we calculate how many pixels are filled by the found SVO node. This operation is performed in the same way for the ray tracing rendering approach and triangle based rasterization.

5.2.3 Optimizations and restrictions

The proposed solution requires an additional rendering step for finding the accurate pattern node. In some cases, this might be too expensive in order to fit the defined time requirements. We must render the 3D objects twice. In order to optimize our algorithm, we can use the previously rendered frame. In that case, apart from rendering the final image, we need to save the depth and voxel id data to an additional render target. After that, using the obtained minimum depth value and the node id stored in the shader storage buffer object, we can calculate the fill rate value by means of a compute shader.

The described algorithms give us the accurate results only when all voxels of a 3D object have exactly the same size. Otherwise, it is necessary to perform an additional calculation to obtain the proper pattern voxel. In order to obtain the correct interpolation weight, which will be used in the LOD transition stage, we must take into account the render target resolution.

5.3 Function boundary conditions

Last but not least an important part of the LOD evaluation algorithm is the boundary conditions of the LOD evaluation function. The defined evaluation boundaries are as follows:

- **Minimum condition** — defines the object minimum LOD. If the rendering of some object on the virtual screen produces N pixels corresponding to the minimum condition, there is no need to execute the evaluation and transition algorithm. This is the universal minimum boundary condition that is be used for any kind of voxel visualization algorithm. We defined N as a parameter, but the perfect function should use N defined as 1.
- **Maximum condition** — defines the object maximum LOD based on the position on the scene in relation to the observer. This boundary is very important because it defines when to stop the execution of the LOD evaluation function. It represents the situation when the object is rendered with the highest possible complexity. The maximum boundary condition is reached when exactly one voxel of the object corresponds to exactly one pixel of the resulting image. We can check this condition by comparing the filled pixel number with the voxel number used to render the image. The atomic counter can be used to calculate how many times a SVO node was rendered on the screen. After that, the sum of all used SVO nodes can be calculated.

6 PATTERN NODE BASED EVALUATION FUNCTION

The object's current LOD can be found using the pattern node pixel fill rate as an argument for the level evaluation function. The proposed evaluation method is based on the extended distance based function presented in section 4.1. The main difference is the usage of the pixel fill rate instead of the distance as the evaluation function parameter. Another very important difference is the type of the propagation function. The distance based approach discussed earlier assumed a linear dependence on the distance. In the case of objects that are represented by the SVO, an object rendered with the N-th LOD fills about four times more pixels than the same object rendered with the N-1-th level. Based on this fact we propose the LOD evaluation function described by Eq. 4 and 5.

$$y = max\Big(0.0, min(\frac{fillRate}{\sum_{i=0}^{N-LOD} maxRate * x}, 1.0)\Big) \quad (4)$$

$$y = \begin{cases} < minRate \Rightarrow \text{decrease level} \\ 1.0 \Rightarrow \text{increase level} \\ (0.0, 1.0) \Rightarrow \text{interpolate levels} \end{cases} \quad (5)$$

where:

y	= LOD interpolation weight
$fillRate$	= pattern node pixel fill rate
N	= number of object LODs
LOD	= object current LOD
x	= defined geometric progression value
$maxRate$	= defined max fill rate
$minRate$	= defined min fill rate

An additional step, which is not necessary with the distance-based approach is the calibration of the object's LOD. Before scene rendering, we must calculate the current LOD for all SVO based objects. In order to do that we use the following algorithm:

1. Find the pattern nodes for all LODs.
2. Calculate the pixel fill rate for all found pattern nodes.
3. Find the minimum fill rate value. Neglect values lower than the specified value or equal to zero.
4. Set the object's current LOD to the level with the minimum fill rate value.

As in the case of using the distance-based evaluation function, our algorithm requires the user input for the minimum and maximum fill rate for each object LODs defined with geometric progression.

For rendering results presented in section 8 we used the following parameters: maxRate = 16 pixels, minRate = 1 pixels and x = 1.

7 LOD TRANSITION

The core stage of the LOD management algorithm is the implementation of the current level transition method. The most straightforward solution to change the current LOD is to just stop the ray tracing algorithm at a specified level acquired from the evaluation pass. In the case of using triangle meshes, change object's vertex data buffers and materials. However, it may produce visible model swapping artifacts that adversely affect the perception and immersion of the virtual scene. Thanks to the voxel representation, the proposed algorithm is free from the limitations imposed by the polygons graphic representation.

The LOD control algorithm aims to change two attributes of the 3D object — geometry and material. In the voxel representation, both these attributes are related. With the SVO structure, each node can be divided into the maximum eight new nodes with different attribute values. When changing to a higher LOD, some child nodes may disappear creating changes in the object's geometry. Other nodes will just change the values of their attributes. A similar situation exists when the LOD decreases. Fig. 2 shows how the object geometry and material complexity changes between three LODs.

We can observe that if some node has a full set of childrens it is quite easy to perform the data interpolation between the parent node and child nodes. For example, using linear interpolation. The problem arises when some potential child node is missing, or when a parent has only one child. In order to accomplish the proper level transition, it is necessary to solve these issues. Below we describe the proposed algorithm for the SVO based object geometry and material transitions.

7.1 Object material transition

The new type of an SVO node is introduced. We call it *redundant node* based on its actual meaning for the object representation. The redundant node can be the

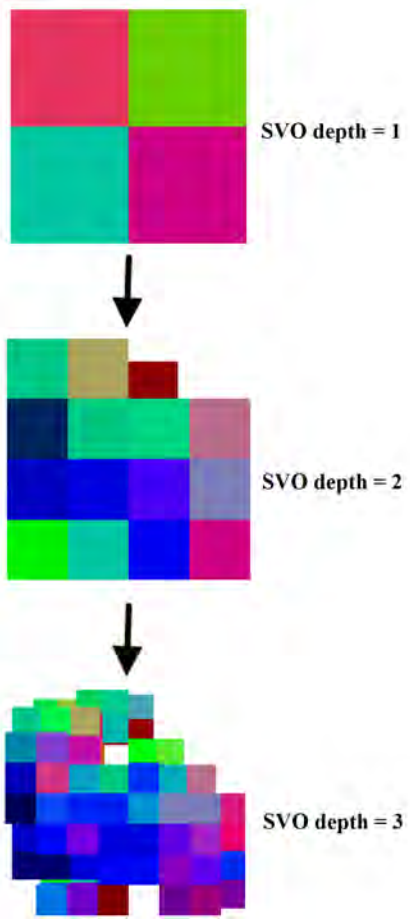


Figure 2: Object differences between three levels of Stanford Bunny [Stanford11].

actual part of the SVO, or it is just an information about the additional node in the tree. If the last traversing node has not the full set of children - we fulfill the gaps with the redundant nodes. We treat the last traversing level of the tree as it has a full set of child's. The main idea behind the redundant node is that in order to perform the interpolation between the parent and the child nodes the number of nodes at both levels must be the same. This is necessary for a continuous transition from one level to another.

All interpolation operations are performed between the parent and child node. Let's start with the parent node representation. As we mentioned before, each node can be divided into maximum eight child nodes. This means that we can treat the parent node as eight identical nodes with the same attributes. The more complex issue is with the children nodes. If the current node does not have eight children we must replace the missing nodes with the redundant nodes. Such nodes will have identical attributes as the parent node. Thanks to that, on both tree levels we will have an identical number of nodes. Fig. 3 presents the idea of using redun-

dant nodes for the two-dimensional grid. For a 3D data structure like the SVO, the method is identical.

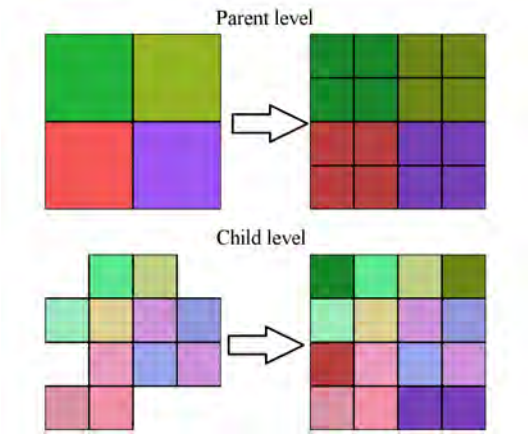


Figure 3: The idea of the dividing a parent node to eight identical nodes with redundant child nodes. Divided parent nodes and redundant nodes are highlighted with stripped lines.

7.2 Object geometry transition

Using the parent node division with the redundant child nodes we achieved the possibility to perform the data interpolation between two LODs. However, the redundant nodes must somehow disappear at the end of the level transition. The most straightforward way to achieve this is by using alpha blending with transparency. Unfortunately, this solution affects other nodes and create unacceptable artifacts.

We propose an alternative solution based on scaling the redundant nodes. Parallel to the data interpolation, with the interpolation weight parameter we change the size of the redundant nodes from the initial values to zero. However, the scale transformation in the defined origin causes the formation of holes at the edges of the objects. Thus, there is the need for an additional redundant node position control, so that they will be absorbed by the nearest neighbor and disappear in a more natural way.

In order to implement redundant nodes fading out we need to find the node's nearest neighbor and calculate the fading direction. The SVO structure guarantees that each node has a connected neighbor. If we cannot find any candidates on the children level, we seek for it at the parent level. The only exception to this rule is the tree root node. The direction vector for the root node will never be required. In the case of the ray tracing approach, we have access to the neighbor nodes during object rendering. If we do not have an access to the neighbor nodes during object rendering, we need to store pre-calculated values in an additional node attribute. Fortunately, we have a finite number of possible directions. A node can have maximum 7 possible neighbors on the node level and 8 possible candidates on the parent level. In order to minimize memory requirements, we can create a lookup table for direction vectors and store only an index to the vector. However, it is only required when our visualization algorithm is not based on the ray tracing approach. Fig. 4 presents an example of performing a continues transition.

Figure 4: Example of level transition based on the developed method.

8 RESULTS

In this section, we present rendering results of the developed algorithm. It is very difficult to present continuous LOD management on static images or even video samples. A fundamental feature of the continuous LOD transition is to hide level changing from the observer. Fig. 5 - 6 demonstrates rendering the result of the developed LOD management algorithm.

9 CONCLUSIONS AND FUTURE WORK

We have developed a novel approach for efficient real-time rendering and controlling the SVO LOD. Our method can be used to algorithmically evaluate the current LOD and perform a transition between two levels. The pixel fill rate method instead of a distance parameter allows for better control of virtual scene rendering results. Moreover, regardless of the chosen rendering method, we propose a universal pattern node evaluation method that can be used in real-time. In the case of the ray tracing approach, the required additional data can be obtained from the rendering pass or based on the previous frame. In the case of the triangle based rasterization method based on the parallel reduce algorithm and GPU queries offers real-time performance.

The LOD transition algorithm allows to perform a dynamic and continues control of the SVO based objects which is our main contribution. By extending the SVO structure with a new type of node called redundant node we achieved the full control of the level interpolation stage. Moreover, the proposed fading algorithm based on the fade out direction and scaling allows for a LOD change without any graphical artifacts or loss of the virtual scene immersion. The developed method is applicable for various voxel rendering algorithms. Moreover, the potential increase of memory consumption for additional data has been minimized.

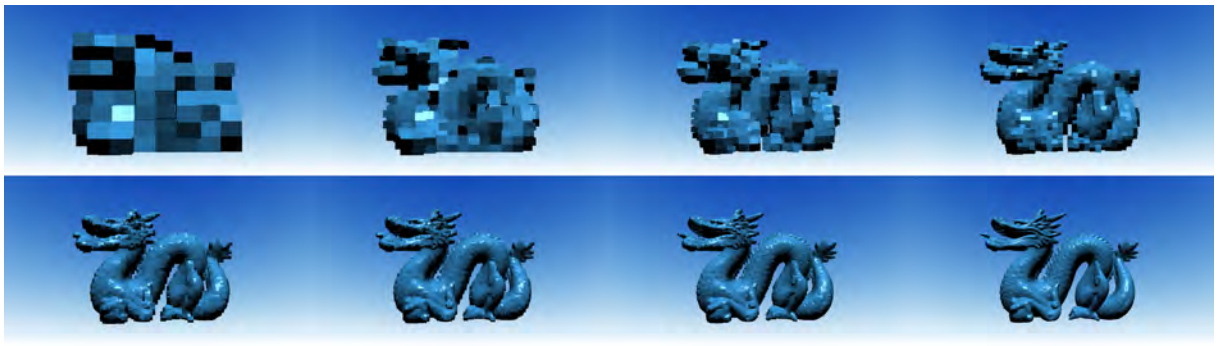


Figure 5: Object geometry and material transition example.

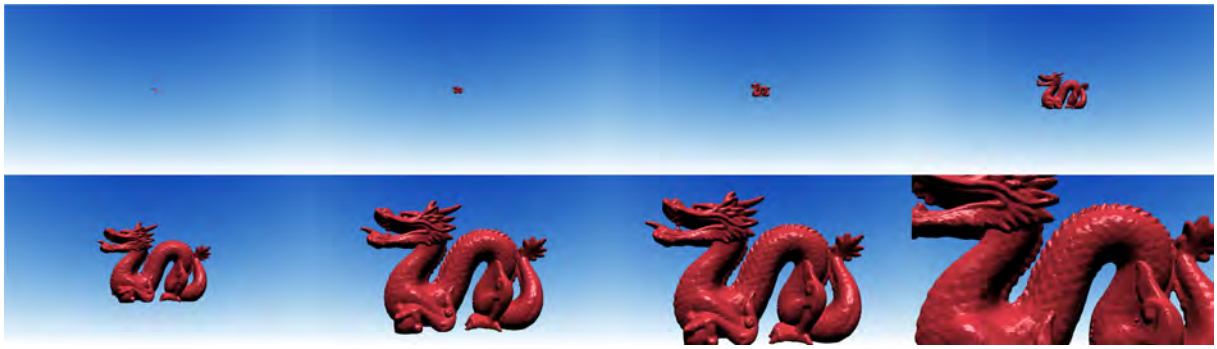


Figure 6: Example of the LOD management algorithm based on pixel fill rate evaluation for the single test object.

An obvious step forward would be to experiment with the control of the entire virtual scene with numerous SVO objects. The current method is dedicated to controlling a per object LOD. Moreover, the proposed method does not perform the LOD evaluation in the view-depended style. We control the whole object details even when we see just part of the object.

Last but not least, an interesting research can be conducted on the situation when we reached the highest LOD of the current object and still could get closer to the object. In that case, the interesting solution seems to be the procedural generation of the geometric complexity using e.g. displacement maps and tessellation.

10 REFERENCES

[Bau11] Bautembach D., Animated sparse voxel octrees, Bachelor Thesis, University of Hamburg, 2011.

[Buck04] Buck, I., and Purcell, T., A Toolkit for Computation on GPUs, In GPU Gems, Addison-Wesley, 2004, pp. 621-636.

[Clark90] Clark, J.H, Seminal graphics. New York, NY, USA: ACM, 1998, ch. Hierarchical Geometric Models for Visible Surface Algorithms, pp. 43-50. [Online]. Available: http://doi.acm.org/10.1145/280811.280921

[Crassin11] Crassin, C., Neyret, F., Sainz, M., Green, S., and Eisemann, E., Interactive indirect illumination using voxel cone tracing, Computer Graphics Forum (Proceedings of Pacific Graphics 2011), vol. 30, no. 7, sep 2011.

[DeCoro07] DeCoro, C., and Tatarchuk, N., Real-time Mesh Simplification Using the GPU. Symposium on Interactive 3D Graphics (I3D) 2007, pp. 6, April 2007.

[Garland97] Garland, M., and Heckbert, P. S. 1997. Surface simplification using quadric error metrics. Proceedings of ACM SIGGRAPH 1997, 209-216.

[Laine10] Laine, S., and Karras, T., Efficient sparse voxel octrees, in Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ser. I3D 2010. New York, NY, USA: ACM, 2010, pp. 55-63.

[Lueb02] Luebke D., Watson B., Cohen, J., D., Reddy, M., and Varshney, A., Level of Detail for 3D Graphics. New York, NY, USA: Elsevier Science Inc., 2002.

[Ramos06] Ramos, F., Chover, M., Ripolles, O., and Granell, C., DGCI, volume 4245 of Lecture Notes in Computer Science, page 460-469. Springer, 2006

[Ripol12] Ripolles, O., Ramos, F., Puig-Centelles, A., and Chover, M., 2012. Real-time tessellation of terrain on graphics hardware. Comput. Geosci. 41 (April 2012), 147-155.

[Rossignac93] Rossignac, J., and Borel, P., 1993. Multi-resolution 3D approximations for rendering complex scenes. Modeling in Computer Graphics:

Methods and Applications (June), 455-465.

[Schaf14] Schäfer, H., Nießner, M., Keinert, B., Stamminger, M., and Loop, C., State of the art report on real-time rendering with hardware tessellation, 2014.

[Schiffner15] Schiffner, D., Stockhausen, C., Ritter, M. Surfaces for Point Clouds using Non-Uniform Grids on the GPU, Short papers proceedings WSCG2015.

[Schroeder92] Schroeder, W. J., Zagre, J. A., and Lorense, W.E.1992. Decimation of triangle meshes. In SIGGRAPH 92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, 65-70.

[Shannon48] Shanno, C., E., A Mathematical Theory of Communication, Bell System Technical Journal 27(3).

[Stanford11] The Stanford 3D Scanning Repository, Stanford University, 22 Dec 2010, Retrieved 17 July 2011.

[Tanen07] Tanenbaum, A. S., Modern Operating Systems, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.

[Wil13] Willcocks, C. G., Sparse volumetric deformation, Ph.D. dissertation, Durham University, 2013.

[Willmott11] Willmott, A., Rapid Simplification of Multi-attribute Meshes, Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics, 2011.

[Wright10] Wright, R., S., Haemel, N., Sellers, G., Lipchak, B., OpenGL SuperBible: Comprehensive Tutorial and Reference, Addison-Wesley Professional, 2010.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Optimized Skin Rendering for Scanned Models

Roger Hernando
ViRVIG Group - UPC,
roger.hernando@gmail.com

Antoni Chica
ViRVIG Group - UPC,
achica@cs.upc.edu

Pere-Pau Vazquez
ViRVIG Group - UPC,
pere.pau@cs.upc.edu

ABSTRACT

Skin is one of the most difficult materials to reproduce in computer graphics, mainly due to two major factors: First, the complexity of the light interactions happening at the subsurface layers of skin, and second, the high sensitivity of our perceptual system to the artificial imperfections commonly appearing in synthetic skin models. Many current approaches mix physically-based algorithms with image-based improvements to achieve realistic skin rendering in realtime. Unfortunately, those algorithms still suffer from artifacts such as halos or incorrect diffusion. Some of these artifacts (e.g. incorrect diffusion) are especially noticeable if the models have not been previously segmented. In this paper we present some extensions to the Separable Subsurface Scattering (SSSS) framework that reduce those artifacts while still maintaining a high framerate. The result is an improved algorithm that achieves high quality rendering for models directly obtained from scanners, not requiring further processing.

Keywords: Skin Rendering, Subsurface Scattering, Physically-based Rendering.

1 INTRODUCTION

There are several factors that distinguish skin from other materials and put it in a very special category. The first one is the complexity of the skin itself, because the skin is made up of multiple layers (epidermis, dermis and subcutis), which are composed of different types of cellular level elements. Hence, they scatter light according to their own composition [12]. The second factor is a perceptual one. Human perception of skin is very accurate. In any rendered scene where a human-like character with visible skin appears, slight errors in its simulation are spotted easier. Imperfections in color or shading easily make the model to look awkward for our perceptual system. Thus, the accurate believable simulation of the subsurface scattering is very important to make the scene convincing. There have been huge advances the last years in the simulation of skin for synthetic imaging. Nowadays, quite realistic effects can be achieved in realtime using screen-space techniques such as SSSS [15]. Unfortunately, the speed comes at some cost, and such fast techniques still suffer from artifacts that become visible if the user zooms in, or if a real image is compared side by side with a synthetic one. Although the perceptual quality of such renderings has been demonstrated previously, there is still room for improvement. Screen-space subsurface algorithms are prone to spreading the filter outside the skin. Although this may be alleviated by some correction factors, still may present itself in the form of halos outside the silhouette. A second artifact appears when the model is not previously segmented (e.g. when it is applied to scanned models), in the form of incorrect diffusion: the algorithm spreads away the skin region thus blurring other elements such as the eyes or the hair. And third, irregular scattering distribution: screen-space techniques do not properly account for the distance, in object space, of the distribution, making the scattering more noticeable in high curvature regions. In this paper we deal with these limitations and propose approximations that solve them, while still maintaining high framerates. Thus, our contributions improve screen-based subsurface skin algorithms in three ways: i) halo removal, ii) limiting diffusion, and iii) curvature-aware scattering. The rest of the paper is organized as follows: Section 2 will review some previous work as well as outline the framework we work upon, Section 3 will detail the improvements of our system, and finally Section 4 will discuss the results and point some lines for future research.

2 RELATED WORK

Subsurface skin rendering/simulation techniques, according to their temporal cost, can be initially classified into off-line or on-line rendering techniques. Off-line techniques are used, for instance, in movies, or in applications which need to compute accurately and in a photorealistic way skin appearance and do not require interactive manipulation. Such techniques involve the accurate simulation of light rays going through the skin simulating their scattering effects, which is a very demanding process in terms of computational time, especially if solved for a high number of ray bounces. In contrast, on-line techniques are useful for real-time environments such as video games, which need real-time interaction and manipulation. The main challenge

of such techniques is to compute an approximation of the complex subsurface scattering effects, which should be good enough to be perceptually plausible, but at the same time fast enough to allow for real-time rendering. Furthermore, they should be easy to implement so that they integrate well with existing pipelines (e.g. rendering engines).

2.1 Off-line techniques

The simulation of scattering inside translucent materials dates back to the radiative transfer equation [2]. Off-line techniques compute the *BSSRDF* accurately, although the full multiple scattering simulation within a *BSSRDF* might be computationally prohibitive. A *BSSRDF* is an 8D function (Equation 1) that describes the light transport from one point to another for a given illumination and viewing direction. Monte Carlo simulation (ray tracing) is often the tool of choice to solve the light transport problem.

Jensen et al. [14, 13] use the complete *BSSRDF* along with a diffusion approximation to model subsurface scattering. The main idea behind this paper is to decouple the incident illumination from the evaluation of the *BSSRDF* by using a two-pass approach. In the first pass they compute the irradiance at selected points on the surface, and in the second pass the diffusion approximation is calculated using the dipole diffusion approximation from pre-computed irradiance samples. The dipole diffusion approximation assumes that the material is homogeneous and semi-infinite, which is not the case of the human skin. This approach is substantially faster than directly sampling the *BSSRDF* since it only evaluates the incident illumination once at a given surface location. Later, Donner and Jensen [8] extended the dipole model into a multipole one, which allows the modeling of multi-layered translucent materials, such as skin. They present a multipole diffusion approximation for light scattering in thin slabs, which generalizes to an arbitrary number of layers. This way, it enables the composition of arbitrary multi-layered materials with different optical parameters for each layer (i.e. roughness and refraction indices). This method is both accurate and efficient, and can be easily integrated into ray-tracing simulation methods using the dipole diffusion approximation to compute the scattering effects.

In a further work, Donner and Jensen introduce a photon diffusion technique to combine photon tracing and the diffusion approximation [9]. This combination enables an efficient render of highly scattering translucent materials while accounting for internal blockers, complex geometry, translucent inter-scattering, and transmission and refraction of light at the boundary causing internal caustics. Instead of sampling lighting at the surface as the previous techniques, this technique performs a photon tracing step to distribute photons in the material and store them volumetrically at the first scattering interaction with the material. Then, the radiant emittance at points on the material surface is computed by hierarchically integrating the diffusion of the light from photons.

More recently, D'Eon and Irving [6] presented a new *BSSRDF* for rendering images of translucent materials. Previous diffusion *BSSRDF*s are limited by the accuracy of classical diffusion theory. However, they introduce a modified diffusion theory which is more accurate for highly absorbing materials near the point of illumination. This new diffusion solution separates single and multiple scattering terms. Moreover, the authors derive an extended-source solution to the multi-layer searchlight problem by quantizing the diffusion Green's function obtaining a quantized-diffusion (QD) model. This can be done because the contribution from many depth sources at once arises from the separability of Gaussian functions. This allows the application of the QD multipole model to material layers several orders of magnitude thinner than previously possible and creates accurate results under high-frequency illumination.

Finally, Habel, Christensen and Jarosz [10] introduce the photon beam diffusion method. Their approach interprets incident light as a continuous beam of photons inside the material. They leverage the improved diffusion model [6], but propose an efficient and numerically stable Monte Carlo integration scheme that gives equivalent results using only 3-5 samples instead of 20-60 Gaussians. This method can account for finite and multi-layer materials, and additionally supports directional incident effects at surfaces. Besides, their numerical approach allows to extend the accuracy and capabilities of the diffusion model and even combine it efficiently with more general Monte Carlo rendering algorithms.

Unfortunately, those methods are not suited for real-time because they require more than a few milliseconds to be computed, limiting the framerate. Moreover, such methods are intended to be used with Monte Carlo rendering algorithms (e.g. path tracing, photon mapping) [20], which definitely are not able to produce high quality noiseless results in real time.

2.2 On-line techniques

On-line techniques are mainly based on, or try to improve, the subsurface scattering by Borshukov and Lewis [1], which approximates subsurface scattering by blurring a 2D diffuse irradiance texture using a gaussian filter. While it is efficient and maps well to the GPU, it neglects the more subtle details of subsurface scattering.

The previous idea is extended by D'Eon and Luebke [7] to develop a high-quality real-time skin shader. The key element is to approximate the multipole diffusion profiles of thin homogeneous slabs [8] of a multi-layered translucent material such as human skin, as a

linear combination of carefully chosen gaussian basis functions, in order to use them to blur the irradiance signal in texture space. Since the gaussian convolution is separable, this allows transforming the expensive 2D convolutions into a cheaper set of 1D convolutions. This representation greatly accelerates the computation of multi-layer profiles and enables improved algorithms for texture-space diffusion and global scattering. In order to compute the light transmitted through thin parts of the object, the technique by Dachsbacher and Stamminger is used [5].

Although the previously mentioned techniques are based on blurring the irradiance signal in texture space providing real-time performance, they scale poorly with the number of translucent objects in the scene, since subsurface-scattering simulation needs to be performed on a per-object basis. To overcome this issue, Jimenez and Gutierrez [15] proposed to translate the simulation from texture to screen space. Diffuse irradiance of all objects is blurred once as a preprocessing step employing sum-of-Gaussians, thereby limiting subsurface scattering computations to visible parts of the objects. Although the algorithm is faster due to the fact that it works on screen space, the algorithm has less information to work with, as opposed to algorithms that work in 3D or texture space. Therefore, the screen-space algorithm loses irradiance in all points of the surface not seen from the camera, since only the visible pixels are rendered. Thus, the method cannot calculate the transmittance of light through thin parts of an object. Moreover, due to this screen space lack of information, the method produces artifacts such as thin halos near the silhouette of the surface. Mikkelsen [18] showed that the surface convolution by a Gaussian function can be weighted with a cross bilateral filter (*CBF*) over an image containing the edges from the observer point of view, thus solving these silhouette errors.

The aforementioned method also fails to simulate light transmitted through high-curvature features because of the lack of lighting information behind objects. For this reason, its authors extended the method to simulate the transmittance of light through skin [16]. They basically propose an approximation to reconstruct the irradiance on the back of an object. This, in turn, is used to approximate the transmittance based on the multipole theory [8]. Such technique requires standard shadow maps as input, which eases its integration with rendering pipelines, also reducing the memory usage compared to previous work techniques which take transmittance into account.

Shah *et al.* [21] propose a method to compute *BSSRDF* using the dipole diffusion model. They employ the dipole diffusion model with a splatting approach to evaluate the integral over the surface area in an image-space framework, in order to compute the illumination due to multiple scattering. The main contribution is to take sample points on the surface visible from the light source, and splat the scattering contribution to all points visible to the viewer within the effective scattering range from each point. Finally, each point on the rendered surface receives the scattering contribution from all points that have an influence on it.

A recent approach by Jimenez *et al.* [17] proposes two real-time models to generate separable approximations of diffuse reflectance profiles to simulate subsurface scattering. It uses just two 1D convolutions, reducing both execution time and memory consumption, while delivering results comparable to techniques with higher cost. To approximate a 2D diffuse reflectance profile by a single separable kernel, the authors relax the requirement of radial symmetry of diffusion models. They also show how by combining importance sampling and jittering strategies (e.g. [11]), a small number of samples per pixel are enough in many cases of practical interest. They use the approach by Jimenez *et al.* to compute the light transmitted through thin parts of the object [16].

Unlike the previous described methods, which are based on gathering the neighboring light in order to simulate the subsurface scattering effects, other authors pre-integrate the effects of scattered light into a texture [19]. They define three regions of the mesh where the subsurface scattering is important to achieve realism: zones with high surface curvature, zones with small surface bumps, and the zones which next to shadow edges. To obtain the scattering that occurs due to the curvature of the surface and the shadow edges, a precomputed subsurface texture is used, and accessed with the surface local curvature and the shadowness level of the region. To take into account the subsurface scattering due the small surface bumps, they propose a strategy of diffuse normals in which they filter the mesh normal map with R/G/B skin profiles. The authors claim that this strategy allows them to achieve non-local effects of subsurface scattering using only locally stored information.

Finally, Chen et al. [3] presented Pre-integrated Deferred Subsurface Scattering (*PDSS*), a technique that adapts pre-integrated skin scattering to screen space, making it suitable for use in a deferred lighting pipeline and increasing its visual quality. Surface curvature is calculated in real time by evaluating the curvature from the gradient of world space normals in the G-Buffer, avoiding curvature calculation artifacts. PDSS has the advantages of being independent of the scene geometry and scaling well in the number of lights and the number of objects. They use the method by Penner and Borshukov [19] to calculate the subsurface scattering, which uses the curvature and a shadowing factor to look up into a pre-baked scattering texture and also the diffused normals. Light transmitted through thin parts of

the object is calculated using the approach by Jimenez *et al.* [16].

3 BACKGROUND

Subsurface scattering is a complex phenomenon which describes how light enters an object, interacts with its different layers, and may exit at various points around the incident point or be transmitted through the object. This effect is described in terms of the *BSSRDF S* which relates the outgoing radiance $L_0(x_0, \overrightarrow{\omega_0})$ at a point x_0 to the radiant flux $\Phi_i(x_i, \overrightarrow{\omega_i})$ at the point x_i from the direction ω_i:

$$dL_0(x_0, \overrightarrow{\omega_0}) = S(x_i, \overrightarrow{\omega_i}; x_0, \overrightarrow{\omega_0}) d\Phi_i(x_i, \overrightarrow{\omega_i}) \quad (1)$$

The subsurface scattering effect can also be described using radially symmetric diffusion profiles. A diffusion profile is a function $R_d(x,y)$ that describes the light reflected around a normally incident pencil beam on the origin of a surface of an infinite half-space. For an homogeneous material, R_d is radially symmetric and can be characterized by a 1D diffusion profile $R_d(r)$, which describes how the light attenuates at each point as a function of the distance $r = ||(x,y)||$ from the incident point. To obtain such diffusion profiles, diffusion theory can be used to reach to a diffusion equation [14]:

$$D\nabla^2\phi(x) = \rho_\alpha(x) - Q_0(x) + 3D\overrightarrow{\nabla} \cdot \overrightarrow{Q}_1(x) \quad (2)$$

where Q_0 is the 0^{th} order source distribution, Q_1 is the 1^{st} order source distribution, D is the diffusion constant, and ρ_α is the absorption coefficient. For an infinite medium this equation has a simple solution, however for a finite media this equation has no analytical solution. Some authors propose techniques to obtain such diffusion profiles numerically [14, 4]. Applying a diffusion profile is simple. Consider a point $P(x,y)$ on the surface. We want to obtain the contribution of all points around P. Part of the light arriving at such adjacent points will penetrate into the object and exit at P, with the specific attenuation given by the diffusion profile $R(r)$, expressed by:

$$M(x,y) = \int\int E(x',y') R_d(r') dx' dy' \quad (3)$$

$M(x,y)$ being the radiant exitance at point P, $E(x,y)$ the irradiance around P, and R_d the diffuse BSSRDF. Equation 3 sums the contribution of each point around P, each of them weighted by the diffusion profile $R(r)$ according to its distance r to P. Therefore, it can be rewritten as a two-dimensional convolution:

$$M(x,y) = E(x,y) * R_d(r) \quad (4)$$

Carrying out the 2D convolution of Equation 4 is costly for real-time applications. However, if $R_d(r)$ can be approximated by a sequence of $2N$ $1D$ separable convolutions, A, represented as:

$$A(r) = \sum_{i=1}^{N} a_i(r) \quad (5)$$

where the approximation A is defined by $1D$ functions a_i. Due to the radial symmetry of R_d the same functions a_i can be employed in both coordinate directions.

3.1 Screen-Space gaussians sum

From Equation 4, D'Eon and Luebke [7] observed that the skin diffusion profile resembles the aspect of a Gaussian, so a sum of Gaussian functions (Table 1) is suitable for approximation, being $R_d(r)$:

$$R_d(r) = \sum_{i=1}^{k} w_i G(v_i, r) \quad (6)$$

Following the previous idea, Jimenez et al. [15] proposed to perform this sum of gaussians approach in screen space instead of texture space. The method requires the diffuse render, the linear depth of the scene, and the stencil buffer to distinguish which zones are skin and which are not. With them, it generates different levels of Gaussian blurring, and adds up all these levels using the weights of Table 1 in order to obtain the subsurface scattering contribution. Finally, it adds up the specular term to obtain the final render. It is worth noting that pixels located far from the camera should have narrower kernel sizes than pixels near the camera, so the width of the kernel should be modified according to the distance to the camera. Besides, a correction component is introduced to prevent scattering through neighboring pixels in screen space but farther away in the geometry.

Variance	Color Weights		
	Red	Green	Blue
0.0064	0.233	0.455	0.69
0.0484	0.1	0.336	0.344
0.187	0.118	0.198	0
0.567	0.113	0.007	0.007
1.99	0.358	0.004	0
7.41	0.078	0	0

Table 1: Sum-of-gaussians parameters for a skin model depicted by D'Eon and Luebke [7].

This way, the technique mimics the results of the method proposed by D'Eon and Luebke [7], at a fraction of its cost both in time and memory usage. What this method can neither reproduce nor match from the previous method is the simulation of transmitted light through the thin slabs of skin. Therefore, this method must be used along with those that simulate forward scattering.

Our technique is based on this approach, but adapting the shaders to handle an arbitrary number of samples.

4 OPTIMIZED SKIN RENDERING

Raw scanned acquired models are 3D photographs of an object, just including color and geometry information, and sometimes a normal map depicting the fine details of the skin (e.g. pores, wrinkles).

In computer games, models are further processed, to identify skin, eyes, and so on. However, in a more general case, such work is not possible, and thus, using the models as is, causes some artifacts or worsen other problems that still appear with the mentioned algorithms. We illustrate some of these problems in Figure 1, namely halos and incorrect diffusion.

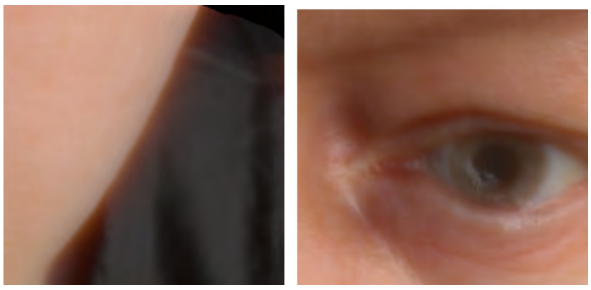


Figure 1: Screen-space subsurface scattering algorithms may produce halos (left) and incorrect diffusion (right) on scanned models.

In this Section we propose some optimizations to address these problems with raw models.

4.1 Halo Removal

The first obvious problem that appears is halos. The screen space approaches produce halos between neighboring zones in image space but at different depth levels. The authors of the aforementioned subsurface scattering methods noticed the halos problems as well, and tried to tackle them with the correction factor (central image of Figure 6). The approach modulates the color of the samples which, although being near the central point of the diffusion profile in image space, are far away in the geometry, using the difference in depth between the central and the sampled points. Unfortunately, the correction factors are not enough and Mikkelsen [18] showed that using a cross bilateral filter (*CBF*) to weight the diffusion profile fixes the halos problem.

A *CBF*, works like a bilateral filter (Equation 9), but uses an auxiliary image to compute the weights instead of the image that is being filtered. *CBF* is characterized by the following equation:

$$CBF[I,E]_p = \frac{\sum_{q\in S} G_{\sigma_s} e^{-||p-q||} G_{\sigma_r} e^{-(E_p-E_q)} I_q}{\sum_{q\in S} G_{\sigma_s} e^{-||p-q||} G_{\sigma_r} e^{-(E_p-E_q)}} \quad (7)$$

where I is the original input image, E is the auxiliary image used to compute the difference of intensities, p are the coordinates of the current pixel to be filtered, S is the window centered in p, and G_{σ_r} and G_{σ_s} are the distance and color weighting factors, respectively.

The auxiliary image E is defined as an image that distinguishes between zones whose normal is perpendicular to the view direction and zones which are not. This creates an image of contours from the point of view of the observer, highlighting the edges between continuous areas in screen space but not in the geometry. This image is defined as follows:

$$E(p) = I(x(p)) * cos^3(\phi_i) \frac{||x(p)||^2}{cos(\phi_j)} \quad (8)$$

where $x(p)$ is the object point which is drawn in pixel p, ϕ_i is the angle between z-axis and the direction from the observer to the point $x(p)$, ϕ_j is the angle between the surface normal and the vector from $x(p)$ to the observer, and $I(x(p))$ the intensity of the pixel p.

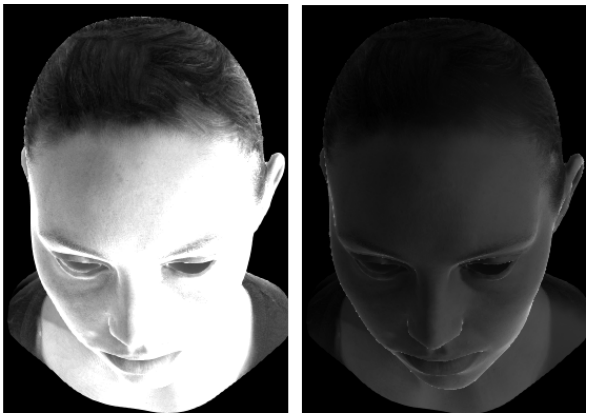


Figure 2: The image shows the unnormalized strategy proposed by Mikkelsen [18] (left) vs. our normalized strategy (right).

Figure 3: The image shows the artifacts (black dots near the edges) introduced by the *CBF* method when used directly with our shaders.

However, the efficacy of this approach as proposed depends on how far the model is from the projection plane, losing the power of detecting edges and therefore not removing the halos, as can be seen in Figure 2. We

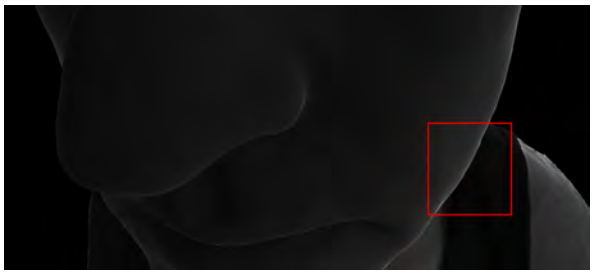


Figure 4: The auxiliary image used in the *CBF* weighting, in order to reduce the halos. The highlighted section corresponds to the region used in Figure 5.

refine this method by modifying the way the auxiliary image is computed. In our case, we make it invariant to the distance by taking the point $x(p)$ as if it was always placed on the projection plane, we call this one the *normalized* image. Moreover, using this cross bilateral weighting directly within our shaders produces some ugly artifacts as can be seen in Figure 3. We fix these issues by modifying the shaders so that if the final color is black, the original diffuse color is used. In contrast to the proposal by Mikkelsen, we use this technique along with the correction factors.

Figure 4 shows the auxiliary image used by the *CBF* with a highlighted area which is the same as used in the Figure 5, which shows the halos effect and its reduction using this method.

Figure 5: Halos comparison: not using any method to correct them (left), using the correction factors (center), and using **our modified *CBF* approach** (right).

4.2 Limiting scattering diffusion

When scanned models are not artist-processed, and thus skin and non-skin areas are not properly segmented, the screen-space subsurface scattering algorithms generate a second artifact: incorrect diffusion. Since the boundaries of the elements are not identified, the method produces blurring over surfaces such as the eyes, as illustrated in Figure 1-right.

We alleviate this problem with a bilateral filter weighted by the color distance of neighboring pixels, which modulates the contribution of each sample. To take into account human perception, the color distance is performed in CIE Lab color space. The bilateral filter we used is defined by the following equation:

$$BF[I]_p = \frac{\sum_{q \in S} G_{\sigma_s} e^{-||p-q||} G_{\sigma_s} e^{-(I_p - I_q)} I_p}{\sum_{q \in S} G_{\sigma_s} e^{-||p-q||} G_{\sigma_s} e^{-(c_p - c_q)}} \quad (9)$$

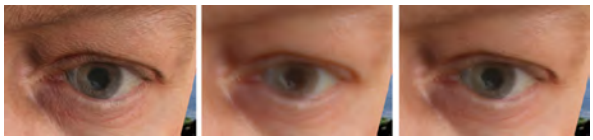


Figure 6: Reducing blur between skin and non-skin zones: without subsurface scattering (left), simple subsurface scattering (centre), and using a bilateral filter to avoid blurring between skin and non-skin zones (right).

where I is the original input image, p are the coordinates of the current pixel to be filtered, S is the window centered in p, I_p and I_q are the colors of the image I at pixel p and q respectively, and G_{σ_r} and G_{σ_s} are the distance and color weighting factors, respectively. As for the *CBF* method, we used G_{σ_r} as the diffusion kernel weight and G_{σ_s} is set to one.

Figure 6 shows the blurring of skin and non-skin zones, and how the bilateral filtering deals with the problem. It is not a perfect solution because it still blurs some high frequency details (i.e. thin hair), but it substantially improves the render quality.

4.3 Scattering modulation

As already stated, scattering is caused by the light entering the surface, bouncing several times, and getting out of it at a different point. If the surface is curved, there will be more light entering and exiting the object. This should be reflected as an increase in the subsurface scattering in higher curvature regions [19]. Unfortunately, the screen-space algorithms presented so far, do not use this information to modulate the amount of scattering. To solve this we modulate the scattering in screen-space so that the effect is stronger at zones with higher curvature and weaker at lower curvature zones.

Like the rest of our method, we are going to compute this in screen-space. To obtain the oriented gradient of a pixel, we use the normals of the neighbors, and obtain the curvature by analyzing the magnitude of the variation of these axes. It is worth noting that, in order not to introduce high frequency discontinuities, the normals used to compute the curvature are the geometry normals and not the normal map normals (Figure 7-left). Besides, the curvature should be smoothed (i.e. mean blurring) to avoid such artifacts.

We have modulated the subsurface scattering effect with the curvature in three different ways (Figure 8):

- Increasing the subsurface scattering strength of a pixel according to its local curvature. However, since screen space curvature is higher at the contours of the geometry, this causes the subsurface scattering effect to be stronger on the edges. Therefore, increasing the filter size at the contours and making the halo artifacts more noticeable.
- Reducing the subsurface scattering effect at zones with lower curvature and keeping it at zones with

Figure 7: Using the normal map to compute the screen space curvature results in a noisier curvature (left). Geometry normals reduce the noise (center). We smooth this map to feed the algorithm with a smoother curvature map (right) free of high frequency discontinuities.

higher curvature. This strategy caused the zones with nearly zero curvature not to simulate the subsurface scattering at all, and breaking the high quality skin rendering.

- Reducing the subsurface scattering effect at zones with lower curvature up to a minimum and leaving it unchanged at zones with higher curvature. This proved to be a good strategy because the subsurface scattering is simulated and strengthens the effect at the high curvature zones.

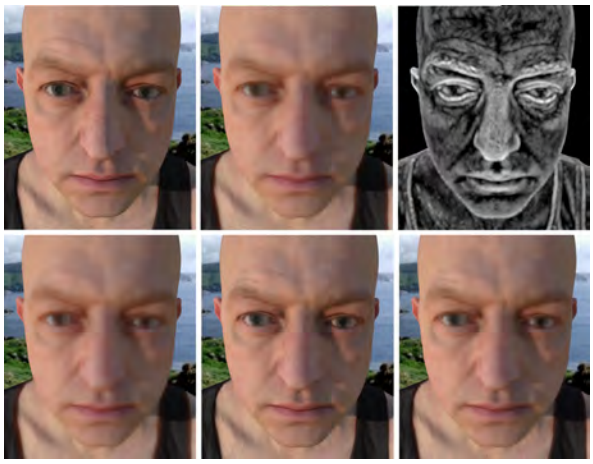


Figure 8: Top row shows a face without and with subsurface scattering. Top right shows the blurred screen space curvature used to modulate the scattering strength. The bottom row shows the three attempted strategies: increasing the subsurface scattering strength according to its local curvature (bottom left), reducing the subsurface scattering effect at zones with lower curvature (middle botom), and reducing the subsurface scattering effect at zones with lower curvature up to a minimum (bottom right).

We have also tried to modulate the forward scattering strength according to the mesh local curvature, which proved to be a bad idea since it produces ugly artifacts (i.e. extremely bright translucency areas) at high curvature zones as shown in Figure 9, where the high curvature areas suffer from artifacts.

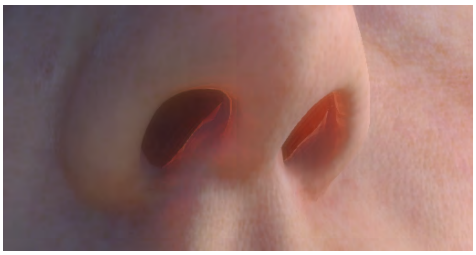


Figure 9: When the forward scattering is modulated with the screen space curvature, it produces bright artifacts at high curvature areas (e.g. nostrils).

5 RESULTS AND CONCLUSIONS

Our implementation also implements other features such as forward scattering to simulate light sources illuminating from back of the object, or gamma correction. The pipeline of our application is shown in Figure 10. In the first step, we get a shadowmap from the light source position. Then, a rendering stage generates the information required for the subsurface scattering: a diffuse map, the stencil buffer, a depth map with linear depth, a specular map, and a curvature map. Then, the rendering stage is the one that generates the subsurface scattering visualization. Finally, a simple step combines the previous result with the specular lighting, and a final tone mapping step generates the final result. Although we work upon the screen-space subsurface scattering work by Jimenez *et al.* [17], our optimizations can also be used on other screen-space methods. The algorithm runs in realtime. The most costly part is the Gaussian sum, which amounts to less than 10 ms for a close view of the face, as shown in Table 2. The remaining steps (shadow map, main render, specular, and tone mapping) add a total of less than 3 ms to the subsurface algorithm.

View	Gausian sum	Artistic	Pre-int Kernel
Close	9.428 ms	1.731 ms	1.799 ms
Mid	2.01 ms	0.492 ms	0.487 ms
Far	0.676 ms	0.312ms	0.36 ms

Table 2: Elapsed time of each subsurface scattering simulation algorithm, at different distances.

To sum up, we have presented a number of optimizations that improve the quality of the screen-space subsurface scattering algorithm: *i)* a technique to avoid halos spreading on different depth regions, *ii)* a method to reduce the scattering diffusion, and *iii)* an improvement tailored to increase the scattering in high curvature regions. The first and third improvements can be applied to any kind of models, while the second is especially suitable for general scanned models that have not been segmented to identify skin and other elements. All of these improvements have a low impact on rendering and thus we have realtime framerates. In future we

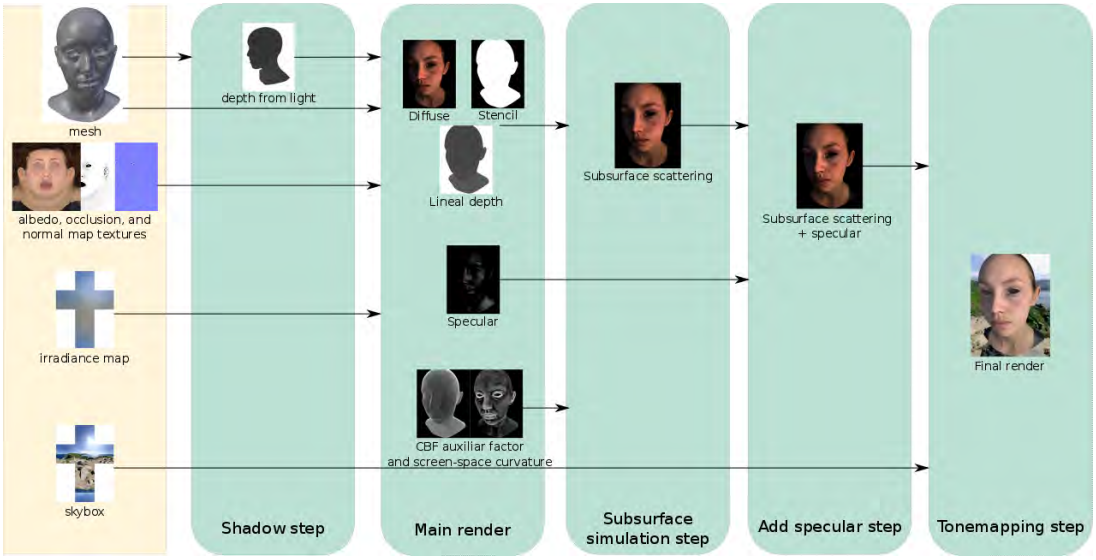


Figure 10: Pipeline of our application.

want to improve the screen-space subsurface scattering approach to better represent the different layers of skin.

ACKNOWLEDGEMENTS

Supported by project TIN2014-52211-C2-1-R by the Spanish Ministerio de EconomÃa y Competitividad with EU FEDER funds.

REFERENCES

[1] G. Borshukov and J. P. Lewis. Realistic human face rendering for "the matrix reloaded". In *ACM SIGGRAPH 2003 Sketches &Amp; Applications*, SIGGRAPH '03, pages 1–1, New York, NY, USA, 2003. ACM.

[2] S. Chandrasekhar. *Radiative Transfer*. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960.

[3] X.M. Chen, T. Lambert, and E. Penner. Pre-integrated deferred subsurface scattering. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 98:1–98:1, New York, NY, USA, 2014. ACM.

[4] Craig D. and H.W. Jensen. A spectral bssrdf for shading human skin. In *Rendering Techniques*, EGSR '06, pages 409–417, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[5] C. Dachsbacher and M. Stamminger. Translucent shadow maps. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 197–201, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[6] E. D'Eon and G. Irving. A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.*, 30(4):56:1–56:14, July 2011.

[7] E. d'Eon and D. Luebke. Advanced techniques for realistic real-time skin rendering. In Hubert Nguyen, editor, *GPU Gems 3*, pages 293–347. Addison-Wesley, 2008.

[8] C. Donner and H.W. Jensen. Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.*, 24(3):1032–1039, July 2005.

[9] C. Donner and H.W. Jensen. Rendering translucent materials using photon diffusion. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, pages 4:1–4:9, New York, NY, USA, 2008. ACM.

[10] R. Habel, P.H. Christensen, and W. Jarosz. Photon beam diffusion: A hybrid monte carlo method for subsurface scattering. *Computer Graphics Forum (Proceedings of EGSR)*, 32(4), June 2013.

[11] J. Huang, T. Boubekeur, T. Ritschel, M. Holländer, and E. Eisemann. Separable approximation of ambient occlusion. In *Eurographics 2011 - Short papers*, 2011.

[12] T. Igarashi, K. Nishino, and S. K. Nayar. The appearance of human skin: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(1):1–95, 2007.

[13] H.W. Jensen and J. Buhler. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.*, 21(3):576–581, July 2002.

[14] H.W. Jensen, S.R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA, 2001. ACM.

[15] J. Jimenez and D. Gutierrez. *GPU Pro: Advanced Rendering Techniques*, chapter Screen-Space Subsurface Scattering, pages 335–351. AK Peters Ltd., 2010.

[16] J. Jimenez, D. Whelan, V. Sundstedt, and D. Gutierrez. Real-time realistic skin translucency. *IEEE Computer Graphics and Applications*, 30(4):32–41, 2010.

[17] J. Jimenez, K. Zsolnai, A. Jarabo, C. Freude, T. Auzinger, X.C. Wu, J. v.d. Pahlen, M. Wimmer, and D. Gutierrez. Separable subsurface scattering. *Computer Graphics Forum*, pages n/a–n/a, 2015.

[18] M.S. Mikkelsen. Skin rendering by pseudo–separable cross bilateral filtering. *Naughty Dog Inc*, page 1, 2010.

[19] E. Penner and G. Borshukov. *GPU Pro 2: Advanced Rendering Techniques.*, chapter Pre-Integrated Skin Shading, pages 41–55. AK Peters Ltd., 2010.

[20] M. Pharr and G. Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.

[21] M. A. Shah, J. Konttinen, and S. Pattanaik. Image-space subsurface scattering for interactive rendering of deformable translucent objects. *IEEE Comput. Graph. Appl.*, 29(1):66–78, January 2009.

Tiber Valley Virtual Museum: user experience evaluation in the National Etruscan Museum of Villa Giulia

Eva Pietroni, Alfonsina Pagano
CNR ITABC
via Salaria km 29,300
00015 Monterotondo-Rome Italy
"eva.pietroni", "alfonsina.pagano"@itabc.cnr.it

Caterina Poli
Tuscia University
Largo dell'Università s.n.c.
01100 Viterbo, Italy
policat@tiscali.it

ABSTRACT

The paper presents a survey on the user experience related to the *Virtual Museum of the Tiber Valley,* an innovative VR installation requiring gesture-based interaction, designed and developed by CNR ITABC and permanently accessible at the National Etruscan Museum of Villa Giulia in Rome. This research arises from the desire of the authors to verify attractiveness, usability, and communication effectiveness of the system with the end users while having such a multi sensorial experience in the museum. The employed strategy in the survey and the final results will be discussed in comparison with authors' expectations, outlining best practices out of this massive study.

Keywords

Virtual Museums, user experience evaluation, gesture-based interaction, virtual reality, emotional storytelling, qualitative and quantitative analyses.

1. INTRODUCTION

Virtual Museums (VMs) have seen a rapid growth in the past years given the big effort in producing always more user-friendly systems, focused on a fruitful contamination among narratives, new interaction paradigms and sensory immersion in 3D environments. Recently the V-Must project has consolidated the idea that virtual museums have not to be considered as simple digital reproduction of physical museums; whereas they need to be conceived as "aggregators" of different contexts and interpretative layers related to the Cultural Heritage, that are not commonly accessible in the real museums, with particular attention to the enhancement of "museum experience through personalization, interactivity and richness of content" (www.v-must.net). Interaction turns to be of utmost importance when we want to make the user really feel involved within the virtual scene.

Thanks to some recent technologies (i.e. motion-capture sensors, head-mounted displays), the *sense of presence* into such VMs has overcome the limits of traditional desktop-based interfaces, opening great perspectives in human-computer interaction, [Syl08]. The chance indeed of simulating the visitor's physical presence inside the cyberspace, by performing body gestures to interact with not tangible contents, represents a great revolution in the way of making experience. Once the user's senses and mind are "embodied" in the artificial environment, he becomes able to perceive his body as part of the virtual scene and then interact with the 3D elements, reaching the sensory immersion in the 3D environment, with greater emotional participation, conceptual engagement and enhanced learning capabilities, [Wit98], [Sla99].

But how can a VM convey such an experience? Visual aspects, narratives and interaction interfaces are of primary importance. Gestures represent our natural language and therefore the most immediate way to interact with the external environment. However the interface elements still play a crucial role in the user's recognition of gestures and movements to be performed when facing natural interaction inside virtual reality applications. The *VMUXE* work, an approach to the user experience evaluation for VMs by CNR ITABC, Fraunhofer Institute and Lund University [Goc13], revealed that the linearity and simplicity of the interface elements highly influence the user's understanding of what to do with VMs and, consequently, the memorization of the interface useful to accomplish the various tasks.

Moreover, the visual information and the language need to be short and accurate so to be understood by a broad audience. This is strengthened also in the *Etruscanning* project (2012) a VR installation by CNR ITABC allowing the public to explore an Etruscan tomb reconstructed in its original aspect, with funerary goods and dead personages inside. The tutorial proposed at the begin of this natural interaction application (composed by figures and few text lines) is welcomed by the users, but it is claimed to have too much text in the explanation [Pie13]. Instead, during the exploration, storytelling in first person and soundscape have been the most appreciated aspects by the public. The evocative reconstruction granted by the system, together with the location inside the Vatican Museums in a dedicated space without distractions, have been underlined to be strength points.

Therefore, not only the graphics but also the interface layout and the stories support the user, by providing an involving and profitable experience. In this sense, the usability evaluation of *Imago Bononiae* project, by CNR ITABC [Fan15b], confirmed that minimal and explicit visual indicators again help users in the recognition of the the urban landscape, allowing them to autonomously "move" inside the 3D scenes. The intense and immersive experience brought the majority of the sample to affirm to be stimulated to know more about the subject of the VM, by visiting the real city of Bologna or deepening the related information on the Web.

The importance of the narrative sphere is proved by Antonaci et al. who presented a pedagogical study on the "Keys2Rome" exhibition (keys2rome.eu), [Ant15], discussing about a natural interaction project, *Admotum*. From direct feedback we can state that users don't want to perceive the complexity of the technology and they desire to fully enjoy the storytelling. Technologies need to be "invisible" [Fan15]. The call for digital stories is finally confirmed by the V-Must Poll on "Quality Labels in Museums" conducted in 2014 [Pes15], where it came out that museum's audience thinks that stories and visual information are more important than getting access to "how" the 3D reconstructions have been done, supported by some visitors that do not think that complex interactive systems are more necessary than stories and 3D visualization.

We can indeed assume that digital narratives, interactive experience and sensory immersion strongly affect the attraction of VMs, allowing the users to better remember the provided cultural information. The case-study we are going to present in the next sections is an attractive on-site virtual reality installation, using gesture-based interaction, dedicated to the Tiber river. Stories are told combining different communication paradigms: cinema, theatre, poetry, augmented reality and game-alike strategies, tied together for the first time in the Cultural Heritage field.

2. THE PROJECT

The *Virtual Museum of the Tiber Valley* has been conceived in order to increment and disseminate the knowledge and the affection towards the territory north of Rome, crossed by the Tiber river, an area 40 km long x 60 km wide. It has been developed by CNR-ITABC in collaboration with E.V.O.CA, supported by Arcus S.p.A. and the Italian Ministry of Cultural Heritage. As the VM aims also at stimulating people to visit the real places, an integrated communicative system and multimedia installations have been realized, diffused in local museums and inside more attended and relevant institutions in Rome.

Starting from a cross-disciplinary study and documentation of the territory and of its evolution across time (from 3 million years ago until today), 3D representations at different scales have been realized, from the whole landscape, to specific sites. The Tiber river is told establishing interconnections among geography, geology, archaeology, architecture, botany, history, literature and mythology, art. One of the results of the project is an attractive VR application characterized by gesture-based interaction and an innovative approach in interactive storytelling. It is accessible as permanent installation in the National Etruscan Museum of Villa Giulia Museum, in Rome, since December 2014 (see Fig.1). This installation is the focus of the current paper. The visualization is distributed on three aligned 65" screens, arranged in a semicircle, in order to arouse a feeling of immersion and perceptive involvement. The user migrates among different "avatars" to explore four virtual scenarios:

1. "*On the spirals of the Tiber: the landscape of the origins*": the user can fly, like a bird (using his arms), over an evocative 3D representation of the middle Tiber valley landscape. The topography is rendered with an evocative and visionary style; 3D graphics resemble a game, the sounds have been composed redeploying traditional folk songs and flock bells. Crossing magic circles, the user can travel back in time, activating movies with stories related to: a) the geological and geomorphological evolution of the territory; b) the potential landscape in the VIII-VII century BC and the birth of cities (3D reconstructions).
2. "*The secrets of the river"*: swimming underwater in the deep of the Tiber like a fish, the visitor can experience the memory of the river; he meets fluctuating images, iconographies, sounds, literary fragments taken from ancient and contemporary poets and authors. Literary quotations come out from a multitude of voices. The visitor uses his arms to follow these images/memories. Movements of other fishes are controlled by artificial intelligence and swarm dynamics.
3. "*Mena's story, Volusii's Villa*": the user acts like a man, walking through a possible 3D reconstruction

of the villa in Augustan time. Here he is involved in the dramatic story of the freed slave Mena, an imaginary character but historically plausible. The archaeological and historical context is used as scientific background of this engaging tale. Through gesture-based interaction, the visitor can navigate the space: he can relax following a predefined camera path, along which he can stop in every moment and look around to analyze details of the architecture and decoration (guided tour with limited interaction).
4. "*Here only you can see me. Lucus Feroniae"*: the user walks through the ancient Roman settlement of Lucus Feroniae reconstructed in 3D during Tiberius' and Trajan's time. He follows predefined camera paths but he finds crossroads where he can choose the "direction" to access different stories and places. Real actors (filmed on a green screen) have been integrated in the virtual scene to represent the ancient characters performing their daily activities. Augmented reality solutions have been implemented as, during the exploration, the current archaeological site and its 3D reconstruction are shown in parallel on the three screens.

Figure 1. VR Installation in the National Etruscan Museum in Villa Giulia (Volusii's Villa scenario)

The user can access scenarios in the order he prefers and he can interrupt each experience in every moment, jumping to another one.
In this installation layered narratives, natural interaction interfaces, embodiment and novel approaches in the integration of different media are considered essential for the cultural experience of end users. They are used to let the visitors feel important and crucial, and to involve them also emotionally. One person at a time can guide the system in the interactive area in front of the screens (4m x 4m). The other users (about 15 persons) can watch from the space all around and they can alternate in every moment in the active role. The interaction interface consists in few coloured circles on the floor (replayed on the screen): when the user walks up to cover a circle, the corresponding scenario is loaded. Moving on the bigger yellow circle in the center, the user can use his arms to explore the selected scenario. A blue silhouette of a figure is always present in the bottom right part on the central screen, suggesting the gestures the user has at his disposal to explore the active scenario (Fig.1). Microsoft Kinect (first generation) has been used for motion capture; it doesn't require any calibration and the user is immediately identified and tracked by the system. The application has been developed in Unity3D. For further information on the systems and the scientific background, please refers to Pietroni et al. [Pie13]. Beside, authors suggest having a look to the demo movie at https://vimeo.com/album/3841439/video/129867454.

2.1 The virtual museum in the real museum

The VR application is located at the first floor of Villa Giulia Museum, in a room dedicated to the Faliscans and Capenates (populations living in the middle Tiber Valley before the Romans' conquest). Entering the room, the visitor can see artifacts in the showcases and, beside, he can interact with the installation. People arrive in this place after having crossed dozens of rooms whose collection are mostly organized according to taxonomical criteria. Thus they are often tired and maybe bored. Authors have not conceived and "designed" the virtual museum for this space: its expected final destination was a secluded and dark room in Villa Poniatowksy, exclusively destined to this installation: a perfect environment to favour the concentration, even if more peripheral. However at the end of the project this precondition failed and the present location in Villa Giulia was considered a possible alternative. Nearby the interactive area, two printed panels and a video tutorial running in loop in a small TV, have been put to introduce and support the visitors' experience. Authors suggest watching the tutorial movie at https://vimeo.com/album/3841439/video/127130786.
After the opening of the installation, authors wanted to investigate the efficacy of the installation in the whole context. The survey has been carried on in Summer and Autumn 2015 on a heterogeneous sample of 117 visitors. In sections 2 and 3 results will be presented and discussed.

3. SURVEY

The core content of User Experience (UX) studies is ensuring that individuals find value in what they are using, playing with, experiencing. In order to be perceived as a meaningful "moment", VM projects must be credible, desirable, useful and usable [Bar94],[Kot09], [Mor06]. When facing digital products, and VR environments in particular, users have the chance to immerse themselves into a context of informal learning, where cognitive and sensory-motor processes (i.e. attention, memorization, pattern recognition, enjoyment, performance, embodiment, emotional involvement etc.) take place [Mat09]. With recent development in ICT and new advanced

applications, the need to understand how people react to digital cultural heritage projects, especially museum visitors, is extremely increased.

3.1 Multi-partitioned analysis

After a long experimentation under the V-Must project [Pes15] [Gra15] to find the best strategy to conduct usability and cognitive studies on Virtual Museums, the evaluation of on site installations has been done using three different techniques:

1. **Active and passive questionnaires.** They reveal basic information about:
- Demographic data (gender, age, occupation…), which are essential to understand users' profiles, as these are significant in providing background knowledge for later analysis;
- Notions concerning the user's knowledge or comfortability with the field of new digital technologies and virtual heritage;
- Detailed experience with specific application's case study.

These are fulfilled by the single user, after their experience with the application.

2. **Driven scenarios.** They allow users to test their abilities and prove their attitudes by means of tasks, while raising up spontaneous impressions. This is possible through the usage of "Thinking Aloud" method, which makes users tell the operator whatever comes into their mind in relation with the experience. The tasks to be accomplished are pre-defined and follow a fixed sequence, articulated by the operator. The user is required to solve these tasks and then to evaluate his performance: if it was easy or not, successful concluded or failed, etc.. The outcomes of this guided virtual exploration, are continuously put in comparison with direct observation made by the operator that highlights relevant aspects of users' feedback on the application's usability, content accessibility and overall engagement.

3. **Observations** (made by external operator). They are essential to have an overview of the context of use and the users' general behaviors and attitudes towards the application. A pre-determined list of features to observe are established and put in sequence, in order to have as accurate and equal framework as possible.

This multi-partitioned analysis turns to be greatly useful when investigating both qualitative and quantitative data (see detail in 2.2.1), because it gives an insight on problematics, viewing the issue from different perspectives. This allows the operator to highlight discrepancies in what was told and what was observed, to verify expectations (of authors and users), and to investigate both usability and comprehension. Moreover it allows us to understand if there is a correspondence (and how deep it is) between difficulty of use and frustration/sense of failure, or on the contrary not necessarily difficult of use generates the desire to abandon the experience. From the **open comments** it is also possible to know the general feeling of the public towards the virtual experience proposed inside the real museum.

3.2 Target and Goals

3.2.1 Target

The survey has been conducted on 117 visitors of different ages and technological attitudes. We have investigated groups' dynamic of participation while interacting with the system, to understand if users naturally change role from active participant to passive observer, alternating and cooperating during the experience. The monitoring of both users typologies has been useful to confirm or reject the authors' preliminary suppositions: passive users generally pay more attention to the content while active ones could probably be more focused on how the system works and how to interact with it.

3.2.2 Goals

The goals of this survey have been (a) to firstly test the attractiveness of the installation, the usability of the system, its main interface features and interaction modes; in parallel, (b) its educational potential. Specifically, for the former, we have analyzed the behaviours of people entering the room, basically if they were immediately attracted by the virtual contents (and by which aspect in particular) and successively by the showcases containing the real artefacts, or the contrary. Regarding the usability, we have examined the interaction between the user and the system to see if the interface elements and the required gestures are able to facilitate the exploration of the installation. We have also analyzed the time of usage and the information accessibility, whether it is easy to go through them or not. We finally have tried to retrace the mental processes that led users to navigate the virtual museum, reaching a satisfying experience. Furthermore, as mentioned in section 1.2, one of the secondary goals is the analysis of pertinence of the exhibition spaces according to the virtual experience modalities.

3.3 Strategy

3.3.1 Qualitative vs. quantitative analysis

The collection of meaningful data sees the combination of two investigative strategies:

1. Quantitative data retrieval. This method is effective to obtain a large number of information units. A statistical analysis should be made possible. We can retrieve quantitative information thanks to:
- multiple-choice questions
- yes-or-no questions
- scales (i.e. give value between 1 and 5….)

2. As quantitative data are often not adequate as a stand-alone evaluation method for the achievement of interpretable results, **qualitative data retrieval** is planned. By using this method, the reasons "behind" the quantitative data should be identified with the purpose of obtaining a better understanding of the

users' reactions and deduce some suggestions for future improvements of VMs. We can retrieve qualitative information thanks to:

- open questions
- free comments
- "other" blank space

Often these strategies investigate similar aspects and they are deliberately repetitive, with some variations and in-depth analyses, so that the user's responses can be properly verified. This is important to understand responses' level of reliability, the easiness of reply, and the users' feelings towards specific themes.

3.3.2 Working plan

The survey has been done over the course of 16 days during the summer. End users have been interviewed alternating between normal working days and festivities, in different hours of the day. In this way we have obtained a representative and heterogeneous sample. Promotional days have been organised too, inviting people to come to the museum and become a "tester" of the application.

Survey information have been mainly collected using traditional paper questionnaires - as this has allowed us to reach several users at the same time. In some cases, we have also adopted questionnaires running on iPad - for its practicality and technological flexibility.

3.4 Analysis and interpretation of data

3.4.1 Demographic data

Of 117 observed users, 44 visitors have been involved in the driven-scenario, 46 answered the active questionnaire and 40 the passive questionnaire. Out of 117 global individuals, 107 interviewed left their personal data, even if in anonymous form: the majority has been women (60%), mostly coming from Europe, 68% from Italy. The users' age is homogeneously spread with a pick of 22% between 40 and 50 years old, followed by 18% between 20 and 30. A very low percentage of younger has been registered, although the innovativeness of the VM. This datum can be easily explained by noticing that the average age is in line with the usual museum visitors' demographics.

3.4.2 Installation's attractiveness

When visitors stop in the Falisci and Capenati's room the attractiveness of the system compared to the traditional showcases is confirmed. According to the **observation,** out of 117 active and passive users, 75% have been attracted exclusively by the installation and did not stop to look at the other objects on display. Users recognise the space as being an interactive area, indeed 63% actively participate by controlling the system, while 34% observe another visitor (only 3% do not stop in the area at all).

From 86 global **questionnaires**, 28% of people says to have been attracted by the graphics in the scenes, the color and the atmosphere. 26% have stopped because they consider the installation to be an unusual thing in a museum and 21% because of a personal interest for the subject (see Fig.2).

From 116 **observations** made, it emerges that 76% do not see the poster explaining the project and 81% neither the small TV displaying the video tutorial.

Questionnaire results confirm that 74% do not notice the video tutorial, while there is a moderate difference regarding the poster - which is not noticed by 51%. This divergence can be explained referring to different factors such as a margin of error in conducting observations, unreliable or uncertain answers from users (6% of the cases). In the end, there is the possibility that the user is not answering honestly or refers to another poster.

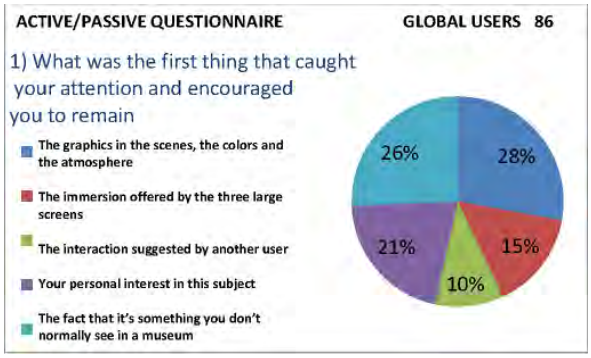


Figure 2. Attractiveness of the system

3.4.3 Usability

From 73 **observations** of active people, it emerges that 93% understand how to use gestures to interact with the system once they enter the interactive area, even if 27% have problems with the gestures and they try to ask suggestion to the operator. The operator provides minimum prompts to the user in the following cases: if the user does the appropriate gestures but is not on the yellow circle; if he moves but does not manage to understand how to interact and he seems near to leave the installation. The first attempt is to suggest the user to examine panels and video-tutorial and then try again.

During the **observation** it has been discovered that **at the very beginning** of the interaction, gestures are performed in different ways: 92% understand the blue silhouette of figure, 45% of them perform the gestures correctly; 47% reproduce the suggested movements but not in a efficient manner; only 8% have difficulty in interacting with the system at all. From the comments collected during the survey, it has been possible to single out the reasons why users could not perform the gestures correctly: 35% do not concentrate or they think it is a game, while 50% do not follow exactly the suggestions of the blue figure.

For the other users that follow the gestures correctly (45%), it emerges that 8% manage to synchronize exactly and replicate every gesture, instead of performing freely the gesture grammar suggested by the blue figure (as it should be done) (see Fig.3).

From a conceptual point of view, this result can be explained by saying that gestures are simulated by a virtual avatar in a continuous sequence but without giving information about the chance to freely use them. Moreover, each of the four scenarios implies different gestures to interact with the system, in fact the user migrates from an avatar to another one and this can be perceived as an amusing factor but sometimes, maybe, misleading.

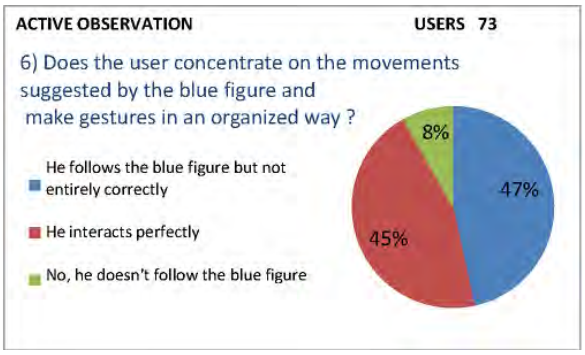


Figure 3. Usability related to gestures

Nevertheless, from **questionnaires,** it emerges that the function of the blue figure is understood in 91% of the cases, and 81% of global users answer correctly about its function, replying that "it is a guide that suggests movements". 5% of users say erroneously that it is "a mirror of yourself" (exclusively passive users) and 7% claim to not understanding its function whatsoever. 2% do not answer.

Of the 46 active **questionnaires**, 50% register an initial difficulty but after a while, throughout the interaction with the system, they are able to comprehend how it works.

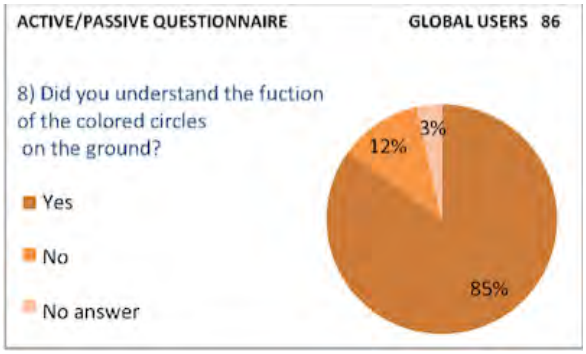


Figure 4. Usability related to the floor interface

For 35% the tasks to accomplish to access information or change scenarios are simple, while for 15% they are difficult from beginning.

From **questionnaires**, it is confirmed that the graphic interface of the floor and the screen is quite well understood. Out of 86 global visitors, 85% understand the function of the colored circles on the floor and 51% specify, correctly, that they are used to change scenes. 12% do not understand their functions and how to perform them (see Fig.4).

3.4.4 Dynamics of participation

From the **notes** taken during **observation and questionnaires** the dynamics of participation can be identified: of 117 observed users, 30% are couples that change between active and passive roles, 31% are groups of 3 up to 7 people and 36% are single persons; only 3% are cases in which entire family or a couple interact with the system playing just one person. In all other cases there is an exchange of role between active and passive experience. There are two questions in the active and passive questionnaires which help understanding the motivations that encourage user to be either an active participant or an observer. In the former, 61% of active users have seen someone else interacting with the system and in 82% of the cases they prefer to be the protagonist, as opposed to 18% that prefer to be the observer. 85% of passive users says they would have liked to interact but they did not. 50% have different reasons i.e. factors external to the application like a lack of time, or the area was occupied by other visitors. Only 7% says that they are not interested and not attracted by the system.

Slight differences appear when comparing the two types of the questionnaires: 5 active visitors prefer the archaeological and historical scenes with 30% of preferences with Villa of Volusii and 28% with Lucus Feroniae; the passive users prefer the symbolic and evocative scenes, where 32% of users vote for the flying scene and 27% for the underwater scene.

3.4.5 Satisfaction

From data collected by **active observation** we see an homogeneous percentages about the duration of user experience. Out of 73 visitors, 28% stay in the interactive area between 15 and 20 minutes, showing a great interest for the VM, independently from the real usability and the final appreciation. It follows a 25% of people who interact between 5 and 10 minutes and 23% between 10 and 15 minutes, with an equal percentage that even overcomes 20 minutes. Only 1% stay for less than 4 minutes. 73% of the active users explore more than one scenario for more than five minutes. This datum is extremely positive given the articulation of the application in multiple scenarios and different levels of learning. This can mean that users are not "afraid" of the interactive

installation, even in cases their performance is not always satisfying.

The virtual worlds explored for a longer time are those of Lucus Feroniae with 49% and the flight scenario with 48%. These are followed by the Villa of Volusii with 36% and the underwater scenario with 30%. It is important to notice that this latter is the scenario with the most de-structured storytelling: here, indeed, the stories do not follow a precise storyline but are poetic fragments floating in the deep of the water. On the contrary, Lucus Feroniae is the most linear scenario, with pre-determined narrative episodes connected by a predefined path. During the visit the user finds some crossroads and he can choose which story to enter. That is why users remain for a longer time: they do not need to perform gestures all the time, the story proceeds anyway, showing the 3D reconstructions and the Feronia's tales deployment. The same goes for the flight scene, which can be interesting and relaxing for users given the particular setting (i.e. blue color predominance; natural gestures…) and the didactic videos that encourage them to stay for a longer time.

In the direct questions in the **questionnaire,** a certain homogeneousness emerges from the quantitative and qualitative data about the appreciation of the different scenes. Out of 86 global users, 26% prefer the flight scenario, 26% the Volusii's Villa, 24% like the underwater scene, 23% the Lucus Feroniae scene and 1% of the users don't answer. From comments left by users we know the reasons why these scenarios are liked. The archaeological historical scenes are liked for the 3D reconstruction of the environment, the site and stories from the characters. The flight scene and underwater scene were liked because of the sensation it created by connecting the way of interacting with the graphics.

Out of 86 global visitors, 44% indicated that the aspects of the system they preferred mostly are the graphics, the colours and the atmosphere and for 39% the exploration based on body movements. Scripts, music and visualization on the three screens seem to be considered of secondary importance.

Comparing these answers with the comments left by the users it emerges that 90% found the experience to be enjoyable for a series of reasons, because of the given information and because it was a new experience, involving and interactive.

The visitors did not exhaust their interest in the Tiber Valley Museum in one visit, out of 86 global users 88% declared that they would return to Villa Giulia Museum. 98% of the cases they are curious to visit the actual places with 64% preferring to visit the archaeological sites and 31% the villages and natural areas along Tiber Valley.

In the free comments about appreciation, two passive users said:" I liked because it's not passive" and "it makes you participate". Other people say "it's surely a beginning of another way to experience the museum", perfectly matching the intent of developers and the new communication strategy that the museum is undertaking.

Eight people expressed a comparison between the real and the virtual museum. They wrote: "it's something different from ceramic", "it gives more information", "the system permits easy understanding of that on display in the showcase", "it permits the ability to connect the objects in the showcase to the context and it inspires new visits and curiosity about the archeological site" and "it helps to introduce the public to art in an enjoyable way".

Other people said "It's rare to find an application like this in a museum", "Normally the interactive applications are dull, instead this is involving". Two active users declared: "too noisy and complicated", "too interactive, it seems like a videogame"

3.4.6 Educational potential

Attention and recognition

Both the active and the passive **questionnaires** ask the user the recognition of scenes out of a collection of 6 images, 4 effectively corresponding to the installation and 2 false. The 2 false images occur in 9% of the cases.

Another question asks to connect various images of scenarios with the corresponding name. The flight and the underwater scenes are matched correctly with a percentage of 71% and 66%, respectively. The Lucus Feroniae and the Villa of Volusii scenes are correctly matched 51% and 43%, respectively. It has been noted that a lot of users declined to answer this question, about 28%. The highest number of incorrect answers however are for the Villa of Volusii with 27% and 23% for the Lucus Feroniae, while the percentage of incorrect answers in the flight scene is 2% and 5% for the underwater scene. It is important to notice that both Lucus Feroniae and Volusii's Villa are both roman sites with some common elements in the story, so this result can be explained. Moreover, while in the flight and the underwater scenes the users need simply to evocatively move in the 3D space with fragmented information provided, Lucus Feroniae and Villa of Volusii scenes are more focused on a topic and need to be followed carefully to understand the storyline.

Memory

Using a series of multiple-choice questions in the **questionnaire** we examine the memorability of the content: the user is asked to remember information provided by characters they have met or to recognize portions of landscapes or specific architectures. Almost all the answers are good: of 86 global users, 62% answer correctly to the questions about the goddess Feronia, 59% to questions about Lucus Feroniae and 68% about the flight scene, 48% to the Villa of Volusii questions; 30% of users decline to answer any question. Despite the considerable level of evasion, this datum is promising, given the innumerable visual inputs and information provided;

this means that users pay actually attention to each of the four scenarios, memorizing the main graphic elements.

Reasoning
Users manage to orientate themselves within the archaeological scenes, with 75% of active users saying that they have understood the function of the "small dot" on the right screen (see Fig.5 and Fig.1).
Of the same group, 71% specify, correctly, that it is used to "indicate your position within the scene". 20% claim not to understand its function, 5% do not answer. From open questions in **driven-scenarios**, the operator understands that users easily recognize the GUI elements and understand their function. In the flight scene, out of 25 users, 68% understand that they are flying over the Tiber Valley; in the Lucus Feroniae scenario, out of 12 users, 83% understand that they are looking at a real archaeological site which has been reconstructed from the same point of view (see Fig.5). Of these, 16% require prompting - so to say they recognise it only when asked.

Figure 5. Lucus scenario: on the left the real site, in the center the reconstructed site from the same point of view, on the right a perspectival view with the user's position evidenced by a white dot.

4. DISCUSSION

4.1 Strength points of the installation

4.1.1 Emotional experience

The design of the *Virtual Museum of the Tiber Valley* aims at making technologies "warm", supporting the capacity of the cultural (and virtual) heritage to generate a feeling of intimate enjoyment, that could translate in spontaneous actions of participation and visit, founded on an enhanced knowledge.
But how can *emotion, involvement, and participation* strengthen the educational potential of a Virtual Museum? This survey has found some answers: historical and archaeological scenarios have been appreciated mostly for the 3D reconstruction of architectures and gardens and the story of major characters allowing to get in touch with the past historical events; evocative scenarios whereas, like flight and swim, have been appreciated for the accurate "sensation" of being there, the calming and relaxing mood created by the interaction, finally for the equilibrated atmosphere and restful colors of the graphics. All these aspects have pushed museum visitors to pay more attention to the installation, gaining benefits out of the contents, as they affirmed.
In general, the graphics is not only an attractive factor, but a stimulus, a fundamental ingredient to make users being involved in the experience. Contrary to the authors' expectations, passive users show a preference for the aerial and underwater scenarios; instead active users prefer archaeological ones where they are involved with less spontaneous gestures. Maybe this can be explained in the light of the theory of mirror neurons [DiD15]: the observers experience a sensation of pleasure in perceiving others swimming or flying.

4.1.2 Gesture-based interaction

Similarly to other VR installations previously developed for museums, and in line with other evaluations conducted [Pie13], [Pes15], the *Virtual Museum of the Tiber Valley* has demonstrated that gesture-based interaction attracts and involves persons of every age and even not familiar with technological devices, as it just requires to perform simple actions in front of the screen. Looking at the average age we can notice that, although the audience is not digital-native, it is still pushed to try the VM installation, spending between 15 and 20 minutes playing with it, inside a museum. Thus not the single good performance but the overall experience and the kind of contents seem to mostly affect the success of the Tiber River project. Another interesting verification is that among active users we have a major percentage of people over 50 years old. This confirms that the technological barriers seem to not divide young from older audience (as observed also in previous projects using natural interaction). Surprisingly, regarding the appreciation, out of 10 people over 60 years old, 6 affirm that the aspect they preferred mostly is the exploration and the interaction, even if the incorrectness in performing gestures seems to be a bit more frequent in this age range.
No meaningful differences emerge between males and females.
Moreover, even if the interaction has been designed for one user at a time, it encourages cooperation, participation, emulation, thus multiplying the impact. This aspect is revealed by users willing to play with the system another time or in other occasions. The curiosity to see the "unexpected" inside a museum like Villa Giulia, brings people to stop and see what is going on with the virtual reality installation.The survey has shown that in most cases difficulty of use does not correspond with frustration and desire to abandon the experience. There is a sense of embodiment [Var91], [Pie13], playful and aesthetic pleasantness that goes beyond gesture-based interaction, making the experience not frustrating but fun, similar to a game, attractive even if not always simple. In our case, users have indeed affirmed to have a pleasant experience while using the installation, new and involving. Definitively, we can confirm that gesture-based interaction let us expand the potential public of virtual installations inside museums.

4.1.3 Storytelling

Authors of the *Virtual Museum of the Tiber Valley* tried to create an involving storytelling going beyond the traditional paradigms of virtual reality through the inclusion of techniques coming from games, theatre, cinema, augmented reality.

This survey let us understand that the undertaken direction in the virtual museum conception (especially regarding "on site" installations) is well accepted and very promising; results in fact reveal that Lucus Feroniae and Villa of Volusii (where storytelling is more structured and adopts some historically plausible "fictions") have reached great appreciation, especially among active users, stimulating curiosity and interest - observable in the positive level of content memorability. Also "visual moods", image effects, camera behaviours and soundscape help creating "the story" to be brought and re-thought at home, after the museum visit.

Authors have received many requests from schools that are interested to continue this educational program in the real context, to have a direct contact with the remains. Further results are expected in the next future, because the new archaeological museum of Lucus Feroniae has been re-opened on the 23rd of April 2016, together with the site of Volusii's Villa, after a long period of unavailability. The virtual museum is actually working as a vehicle of interest, multiplying the public's expectation for the new opening. A great interest towards both the virtual and the real visits is demonstrated also on Facebook at www.facebook.com/muvivate/?fref=ts.

The future research needs to continue following this approach, bringing screenplay and storytelling more and more powerful and professional, finding a link between the way to tell stories and interaction interfaces - possibly clearer.

4.2 Weaknesses of the installation

4.2.1 Gestures interpretation

Despite the graphic interfaces being rather well understood, an element seems to be a bit problematic: the blue silhouette, bottom right on the central screen, showing the gestures the user has at his disposal to explore the environment. Gestures are shown in a loop sequence and each one lasts few seconds. This choice was done to keep the instruction as simple as possible: a unique silhouette occupying a limited portion of the screen and catching the user's attention. The authors' assumption was that, once understood the gestures, the user would have automatically performed them freely, to follow the desired directions during the exploration. On the contrary the survey has shown that many users synchronize themselves with the silhouette's gestures, rather then feel free. It is interesting to notice that this misunderstanding is much less evident when the user is required to swim and flight: in this cases, being the embodiment more immersive, the users are induced to be free and spontaneous in performing actions. The problem does not exist in case of crossroads, when the poses of the blue silhouette are multiplied on the screen to suggests how to go left, right, forward, (see Fig.6).

Figure 6. Poses of the silhouette on crossroads

4.2.2 Support and tutorials

This survey has shown the importance for the public to be supported by the museum's staff to have a full comprehension of the potentialities of interactive installations, in fact in some cases (generally 10% with peaks of 30-40%) users have reached the goal after a suggestion coming from a physical guide (in this case the operator). In few cases panels or video-tutorial have been noticed and effectively used by visitors to learn how to interact with the system. Users generally prefer the immediate experience to learn, facilitated, in this case, by the easiness of body gestures. In case of failure in the interaction, they prefer to be assisted by a living person. This condition recurs several times in virtual installations proposed in real museums, even if the interface design is simple and minimal. In general, while living an embodied and sensory-motor experience, the public tends to evade the reading of texts on panels (even when such texts are very short) or the GUI on the screens explaining how to use the installation.

4.3 Dynamics of participation

4.3.1 Active and passive users

An interesting result concerns the comparison between active and passive users. Authors supposed that individuals directly using the installation would have been more focused on the interface and gestural aspects, while observers would have payed more attention on stories and contents. Actually, results of the survey tell us the contrary: the former, more concentrated in the interactive area, are perceptively absorbed by storytelling and scenarios; the latter, instead, are a bit more distracted by other museum visitors. 23% responses of questionnaires reveal that the museum setting do not contribute positively in making the users concentrate and vigile on what is going on the three screens.

4.3.2 The role of the museum's personnel

Most of the public has expressed the wishes that future evolution of museums can follow such an approach, to overtake the static nature of actual exhibitions. However results have confirmed that not only the research in the field of virtual museums need

to evolve, but the museum's management as well, and the ability/availability of the museum's personnel to support the public dealing with interactive experience and digital technologies. Technologies represent a great opportunity to transmit culture contents to the public but they cannot be abandoned to themselves. Definitively strong collaboration between researchers, creative talents and museum's curators and personnel is required, aiming at strengthening the central role of the public, as main addressee of the cultural experience.
In the case of Villa Giulia Museum, it could happen that some museum's keepers turned down the volume of the application in order to be not disturbed during their work. This has negatively affected the installation's use.

5. CONCLUSIONS

Recently, psychologists, neuroscientists and philosophers have put in evidence the role of emotions in creative processes and intuitive human knowledge: the knowledge and experience of something always requires the activation of an emotion [Cia01]. Emotions can motivate understanding, self-identification, contributing to higher cognitive process of learning [DiD15]. Therefore they represent a method to easily access the culture for everybody, promoting a greater social inclusion. This is confirmed by several evaluations of the user experience inside virtual museums realized in the past years [Pie13] [Pes15] and by the one presented here: users' main expectation is to *enter and interact inside stories*, personalizing their experience, as if they would have been really there, with an active role. Storytelling, embodiment, evocations are key issues. It is of crucial importance to evaluate how people react to digital contents and interaction approaches proposed by researchers and creatives; it is really difficult to match the expectations of such an heterogeneous audience in museums, but some fundamental criteria making an interactive installation successful are today more and more consolidated.

6. REFERENCES

[Bar94] Barrett, E., Sociomedia, Multimedia, Hypermedia, and the Social Construction of Knowledge. Digital Communication series, The MIT Press, 1994.

[Cia01] Ciarrochi, J., Forgas, J.P.. Mayer, J.D., Emotional intelligence in everyday life, in Psychology Press, Taylor & Francis Group, 2001.

[DiD15] Di Dio, C., Ardizzi, M., Massaro, D., Di Cesare, G., Gilli, G., Marchetti, A., Gallese, V., Human, Nature, Dynamism: The effects of content and movement perception on brain activations during the aesthetic judgment of representational paintings, in Frontiers in Human Neuroscience, 2015.

[Gra15] Graf, H., Keil J., Engelke T., Pagano A., Pescarin S., A Contextualized Educational Museum Experience - Connecting Objects, Places and Themes Through Mobile Virtual Museums. In Proceedings of Digital Heritage 2015, Granada, Ed. IEEE, 2015.

[Kot09] Kotsakis, K., Liarokapis, F., Sylaiou, S., Petros P., Virtual museums, a survey and some issues for consideration. Journal of Cultural Heritage, Vol. 10, 2009, pp. 520-528.

[Mor06] Morganti, F., Riva, G., Conoscenza, comunicazione e tecnologia. Aspetti cognitivi della realtà virtuale. LED ed., 2006.

[Mat09] Matlin, M., Cognition, Holboken, NJ, John Wiley & Sons, Inc., 2009.

[Pie13] Pietroni, E., Palombini, A., Arnoldus H., A., Di Ioia, M., Sanna, V., Tiber Valley Virtual Museum: 3D landscape reconstruction in the Orientalising period, North of Rome. A methodological approach proposal, in Proc. Digital Heritage 2013, Vol. II, IEEE, pp. 223-331.

[Pes15] Pescarin et al., Del. 7.1 Virtual Museum Quality Labels. V-Must.net deliverables' collection, Ed. 2015.

[Pie13] Pietroni, E., Pagano, A., Rufa C., The Etruscanning project: Gesture based interaction and user experience in the virtual reconstruction of the Regolini-Galassi tomb, in Digital Heritage Proceedings 2013, Marseille France, IEE, ISBN: 978-1-4799-3169-9, Vol II pp. 653-660

[Var91] Varela, F., Thompson, E., Rosch, E., The Embodied Mind. Cognitive Science and Human Experience, MIT Press, Cambridge, 1991.

[Ant15] Antonaci, A., Pagano, A., Technology enhanced visit to museums. A case study: Keys to Rome. In proceedings of INTED2015, Madrid, Spain, 2-4 March 2015.

[Goc13] Gockel, B., Eriksson, J., Graf, H., Pagano, A., Pescarin, S., VMUXE, An Approach to User Experience Evaluation for Virtual Museums. In Proceedings "The HCI International 2013", Ed. Springer, Heidelberg.

[Wit98] Witmer, B. G., Singer, M. J., Measuring presence in virtual environments: A presence questionnaire. In "Presence", Vol. 7, No. 3, June 1998, 225-240.

[Sla99] Slater, M. 1999, Measuring presence: A response to the Witmer and Singer Presence Questionnaire. In "Presence", 1999, 8(5), 560-565.

[Syl08] Sylaiou, S., Karoulis, A., Stavropoulos, Y. and Patias, P., Presence-Centered Assessment of Virtual Museums' Technologies. In "Journal of Library and Information Technology", Vol. 28, No. 4, July 2008, pp. 55-62 , DESIDOC.

[Fan15] Fanini, B., et al., Engaging and shared gesture-based interaction for museums the case study of K2R international expo in Rome. In Proceeding of Digital Heritage, 2015, Granada. Vol. 1. IEEE, 2015.

[Fan15b] Fanini, B., and Pagano, A., Interface design for serious game visual strategies the case study of "Imago Bononiae". In Proceeding of Digital Heritage, 2015, Granada. Vol. 2. IEEE, 2015.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Efficient B-spline wavelets based dictionary for depth coding and view rendering

Dorsaf Sebai
Cristal laboratory, ENSI
Tunisia

Faten Chaieb
Cristal laboratory, ENSI
Tunisia

Faouzi Ghorbel
Cristal laboratory, ENSI
Tunisia

ABSTRACT

Video representations that support view synthesis based on depth maps, such as multiview plus depth, have been widely emerged raising interest in efficient depth maps coding tools. In this paper, we propose an innovative sparse decomposition on wavelets based dictionary specially designed for the piece-wise planar nature of depth signal. We also evaluate performances of the proposed dictionary for depth maps coding while paying special attention to the impact of depth coding errors on resulting synthesized images. Obtained results prove the relevance of the proposed scheme able to considerably improve the perceived quality of synthesized images.

Keywords
Depth maps, synthesized images, compression, B-spline wavelets based dictionary.

1. INTRODUCTION

Multiview Video plus Depth (MVD) includes sequences of texture images and their corresponding depth maps. The latter are bi-dimensional gray level images representing the distance of each pixel to capture camera. Recent efforts point toward an efficient coding that preserves depth maps particularities, namely their piece-wise planar conception and the critical impact of pixels near contours on perceptual quality of synthesized views [1].

In this context, many coding research work aim at faithfully reconstruct depth map specific piece-wise planar conception. Morvan et al [2] exploit the linear piece-wise nature of platelet and wedgelet functions to approximate depth planar surfaces separated by shaped edges. The wedgelet representation is retained for 3D High Efficiency Video Coding (3D-HEVC) standard [3]. Maitre et al. [4]. propose a codec that relies on a lifting implementation of Shape-Adaptive Discrete Wavelet Transform (SA-DWT). SA-DWT independently treats surfaces separated by edges which, and unlike classical wavelet transforms, provides much sparser decomposition with small coefficients along depth discontinuities. Furthermore, Shen et al. [5] present a new set of Edge-Adaptive Transform (EAT) as an alternative to the classical Discrete Cosine Transform (DCT). EAT avoids filtering across depth discontinuities and so avoids creating large coefficients. However, transform domains used in [4] [5] need an encoded representation of major edge locations to be shared between both encoder and decoder sides.

Since depth images are used for view synthesis and are not themselves displayed, later efforts aim at reducing depth maps coding artifacts that cause severe distortion of synthesized views. Cheung et al. [6] define ”Don’t Care Regions” (DCRs), for each pixel, where a depth value outside the DCR will lead to a synthesis distortion larger than a threshold value. Then, they perform sparsification of the depth map in an orthogonal basis, optimally trading off its representation sparsity and its adverse effect on synthesized view distortion. More recently, this idea is reused by Cheung et al. [7] replacing DCRs by penalty function. For each pixel, a quadratic penalty function is defined based on sensitivity of interpolated images to pixel depth values during rendering process. Transform domains used in [6] [7] are classical orthogonal basis that represent dictionaries of minimum size, concentrating the signal energy over a set of few vectors. However, vectors sets larger than basis, particularly redundant dictionaries, are needed to build sparse representations of complex signals. In the last few years, the emerging attention is to enlarge common orthogonal bases through the design of suitable redundant dictionaries positioned as an interesting alternative. The latter can be a mixture of orthogonal bases and/or dictionaries. Such merging approach aims to design domains where each sub-dictionary is suit-

able for representing one of the signal components. The approaches for learning dictionaries from large input data sets have also been envisioned in order to enhance the correlation of dictionary atoms to signals. However, learned dictionaries are further sensitive to image variations of practical scenarios. Furthermore, if the learning process of the dictionary cannot be repeated in the decoder side, the dictionary transmission is necessary. Increases in terms of storage expense and codec complexity are also noticed due to the feature-dependent nature of learned transform domains.

In this paper, we are interested in studying a predefined mixed dictionary adapted to depth maps sparse representation. In fact, many efforts were carried out to study the most appropriate dictionary for a given class of images such as astronomical images and cartoon-images. This is not the case for the particular class of depth maps. Being redundant, the proposed dictionary, unlike orthogonal basis used in [6] [7], promotes sparsity and avoids high coefficients mainly near contours. Being predefined, the proposed dictionary, unlike the non-fixed EAT and SA-DWT, does not imply a coding overhead for the transform reconstruction in the decoder side. The proposed dictionary is then exploited for depth maps compression to evaluate its relevance for synthesis quality.

Section 2 brings particular attention to fundamental concepts of sparse representations. In Section 3, we aim at studying an efficient dictionary in terms of depth maps sparsity-distortion tradeoff. The dictionary is then exploited, in Section 4, for compression purpose taking into account the quality of view synthesis process, the ultimate depth maps application.

2. SPARSE REPRESENTATIONS

Classical transform coding techniques make use of orthogonal basis, such as Fourier and cosine basis. In such transform domains, signal representation is unique. More recently, sparse representation concept has been developed and its exploitation in image processing is increasingly expanding. Sparse representations proved their performances for texture images compression. It is therefore interesting to explore them for depth maps compression.

Sparse representations distinguish significant components of a signal as a small number of elementary signals selected from a very large transform domain, named redundant dictionary. Sparse representations aim at finding a representation y of the original signal as a compact linear combination of a small atoms number weighted by transform coefficients : $y = D\ x$ where $y \in \mathbb{R}^M$ the representative vector of the original signal of dimension M and $D \in \mathbb{R}^{M \times n}$ a dictionary of n atoms with $n >> M$. $x \in \mathbb{R}^n$ is a sparse vector of transform coefficients. Sparsity of vector x refers to the number of zero coefficients it contains. Because of dictionary redundancy, signal representation is not unique and several combination of vector x are possible. The most appropriate combination corresponds to the sparsest one, i.e. the one with the fewest non-zero coefficients. The Orthogonal Matching Pursuit (OMP) [8] is one of the most developed decomposition algorithms devoted to search such a combination. OMP is a greedy multistage decomposition algorithm that selects, at each iteration, the most correlated atom to the original signal and then subtracts its contribution. This process is iteratively repeated for the residual signal in order to achieve an approximation tolerating an admissible reconstruction error ρ.

3. DEPTH MAPS SPARSITY-DISTORTION TRADEOFF

Efficiency of depth maps representation, both in terms of sparsity and similarity to original data, highly depends on transform domain choice. It seems useful, even required, to use atoms highly correlated to depth maps that we try to model.

3.1. Discrete Cosine/Linear Discrete B-Spline Wavelets dictionary

As introduced in Section 1, depth maps include two major components, namely smooth regions and depth discontinuities. Then, it is suitable to combine, in the same dictionary, two sets of atoms conducive to each of them. In that way, we guarantee complementarity of concatenated atoms where each type of them is capable of reconstructing some signal characteristics that the other one is unable to efficiently do. Typically, we propose the Discrete Cosine/Linear Discrete B-Spline Wavelets (DC/LDBSW) dictionary that includes two kinds of atoms :

Discrete Cosine (DC) atoms for smooth regions : DC atoms of (1) are stemmed from discrete cosine transform :

$$DC = \left\{ \cos\left(\frac{\Pi(2i-1)(k-1)}{2n} \right), i \in \{1, ..., M\}, k \in \{1, ..., n\} \right\} \quad (1)$$

where M is the signal dimension. The dictionary size n is equal to rM with $r \in \mathbb{N}^*$. If $r = 1$, DC is an orthogonal basis. Otherwise, DC is a dictionary of redundancy r. The DC atoms are indisputably adapted to smooth areas representation. This is even more valid for depth maps where smooth areas do not present texture, such as for natural images, but distances of scene objects to capture cameras.

Linear Discrete B-Spline Wavelets (LDBSW) for depth discontinuities : LDBSW atoms, defined by 2, are translated and discretized versions of linear B-spline wavelets at different resolution levels j. The discretization consists in considering the linear B-spline wavelets values at equally spaced knots on a compact interval with distance $\frac{\mathbb{Z}}{2^{j+1}}$ between two adjacent knots :

$$LDBSW = \left\{\varphi_2(i \quad -k),\ i \in [1,M] \cap \mathbb{Z}\right\} \cup \left\{2^{\frac{j}{2}}\psi_2(2^j i - h),\ i \in [1,M] \cap \frac{\mathbb{Z}}{2^{j+1}}\right\}_{j \in [0, log_2(M)-1] \cap \mathbb{Z}} \quad (2)$$

where $k \in [0, M[\cap\mathbb{Z}$, $h \in [0, 2^j M[\cap\mathbb{Z}$ and

$$\varphi_2(x) = \delta_{x,1}$$

$$\psi_2(x) = \frac{1}{12}\varphi_2(2x) - \frac{1}{2}\varphi_2(2x-1) + \frac{5}{6}\varphi_2(2x-2) - \frac{1}{2}\varphi_2(2x-3) + \frac{1}{12}\varphi_2(2x-4)$$

φ_2 and ψ_2 are scale and wavelet functions. M is the signal dimension and j the resolution level ranging from 0 to $log_2(M) - 1$. k and h are translation parameters of φ_2 and ψ_2, respectively. The cut-off approach is used for translation of φ_2 and ψ_2 at signal interval boundaries. This introduces redundancy by considering all the wavelet functions having non-trivial intersection with the interval. LDBSW atoms are piece-wise linear so they comply with the piece-wise planar definition of depth maps. Furthermore, the particular B-splines wavelets of order 2, compared to higher order B-splines wavelets, greatly limit edge smoothing and model in a better way the sharp depth details. This allows a good quality for view synthesis, the ultimate depth maps application.

To build the bi-dimensional dictionary suitable for image processing, the tensor product of the so constructed unidimensional dictionary with itself is considered.

3.2. Complementary of DC and LDBSW atoms

We consider two (8×8) blocks that respectively present areas with and without contours of *Breakdancers* [9] depth map. The original signal in figure 1 corresponds to the concatenated columns of the block into a single one-dimensional vector of depth values. The residual signal results from few OMP iterations. As shown in figure 1, the residual signal of the smooth block is already uniform and is set around zero. However, residue of the block containing discontinuities is not yet uniform and requires more iterations. We can retrieve the piece-wise linear shape of LDBSW atoms that will be useful to reduce this residual signal in next iterations of OMP.

3.3. Comparison and discussion

In this section, we aim at judging the efficiency of DC/LDBSW dictionary for depth maps sparse representation. As already mentioned, the DC atoms stemmed from discrete cosine transform are well suited for smooth areas approximation. It then remains to assess reliability of LDBSW atoms for depth discontinuities representation. For this purpose, we compare the DC/LDBSW dictionary to Discrete Cosine/Linear Discrete B-Spline (DC/LDBS) dictionary, where LDBS atoms are translated and discretized versions of linear B-spline functions of different supports. We also carry out a comparison to Discrete Cosine/Cubic Discrete B-Spline Wavelets (DC/CDBSW) and Discrete Cosine/Directional Anisotropic Atoms (DC/DAA). As well as LDBSW dictionary, atoms of CDBSW dictionary are stemmed from discrete B-spline wavelets. The difference lays in the mother wavelet order that it is no longer linear. Used B-spline wavelets in CDBSW dictionary are cubic (i.e. order 4). Being the successors of X-lets, atoms of DAA dictionary are 2D non-separable functions built by applying geometric transformations to a generating mother function [10]. The latter is a smooth low resolution function in the direction of the contour, and behaves like a wavelet in the orthogonal direction. Using LDBS, CDBSW and DAA dictionaries for comparison is not randomly made. The latter have proved among the most pertinent for signal sparse representation. Furthermore, comparison to these dictionaries would allow us to stress the relevant properties of LDBSW atoms for depth maps sparse representation.

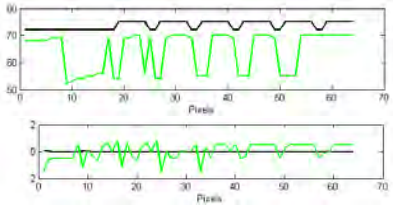


Figure 1. Original signal (top) and residual signal (bottom) issued from few OMP iterations for blocks of *Breakdancers* depth map : smooth block (black) and block with discontinuities (green).

As comparison criterion, we make use of Sparsity Ratio (SR) metric. It is defined as the number of pixels in the image divided by the number of non-zero coefficients used for its representation. A high value of SR reflects the dictionary ability to represent signals with the least number of transform coefficients. Figure 2 presents SR values obtained by sparse decomposition of *Breakdancers*, *Ballet* [9] and *Champagne* sequences on candidate dictionaries using OMP algorithm for different PSNR values.

As shown in figure 2, DC/LDBSW dictionary achieves higher SR values than DC/LDBS. This is thanks to the oscillatory behavior of LDBSW atoms that makes them visually more similar to OMP residual signals than LDBS atoms (*see* figure 1). Compared to DC/CDBSW, DC/LDBSW dictionary allows better sparsity-distortion performances. In fact, LDBSW atoms are B-spline wavelets of lower order than CDBSW ones. This allows them to strongly limit depth discontinuities smoothing, which is crucial for view synthesis.

For 1D signals, wavelets are recognized to be efficient

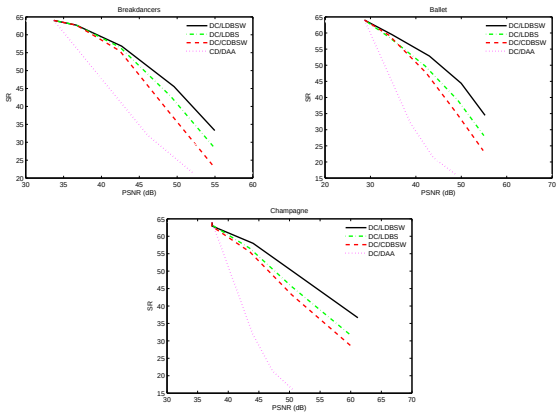


Figure 2. SR values obtained by sparse decomposition of *Breakdancers*, *Ballet* and *Champagne* sequences on DC/LDBSW, DC/LDBS, DC/CDBSW and DC/DAA dictionaries using OMP algorithm for different PSNR values.

for sparse representation of piece-wise smooth singularities. Despite their success, wavelets lose their optimality when extending them to 2D. They fail to detect regularity of contours. In order to overcome the non-optimality of 2D wavelets, it has been proposed to use geometrical-oriented atoms, i.e. the X-lets. Recently, efforts have been made towards redundant dictionaries of transformed generating function using, as DC/DAA, anisotropic geometric transformations. However, geometric atoms relevance for smooth and regular contours of natural images significantly decreases for sharp and irregular discontinuities of depth maps. In fact, DC/LDBSW dictionary achieves, as shown in figure 2, sparser depth maps representation than DC/DAA. This is particularly clear for depth maps with strong discontinuities such as *Ballet* and *Champagne*.

4. SYNTHESIZED VIEWS RATE-DISTORTION TRADEOFF

As a conclusion of the previous section, DC/LDBSW combination allows the best depth maps sparsity-distortion performances against other candidate combinations. This may presage efficient results for depth maps compression. Thus, we integrate DC/LDBSW dictionary within a depth maps compression scheme taking into account the quality of rendered views.

4.1. Compression method

As it has been observed that efficient depth maps compression is achieved by applying a down-sampling prior to encoding [11], the proposed scheme carries out a decimation by a factor of 2 of the initial depth map. One in two pixels is retained per row and per column. Then, an edge detection is applied to decimated depth map. Resulting edge image is next divided into blocks labeled as 1, if they

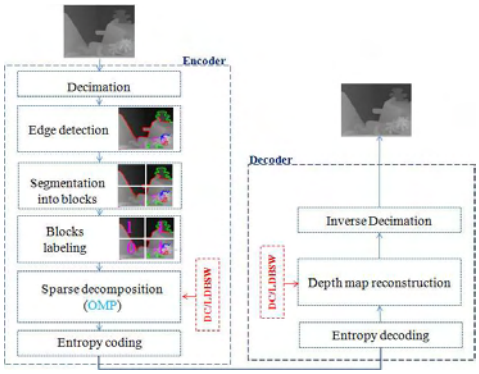


Figure 3. Flowchart of the proposed method.

include contours, and 0 otherwise. Sparse representation of each block is performed using the OMP algorithm on DC/LDBSW dictionary.

As stated in Section 1, coding distortions near contours lead to harmful artifacts of synthesized views. Whereas, coding degradation in smooth surfaces has limited impact on synthesized views quality. Then, we typically adapt the stopping criterion of OMP algorithm to the nature of depth maps blocks, whether they contain contours or not. In order to favor sparsity for smooth blocks (i.e. labelled as 0), the approximation issued from the first iteration of the OMP algorithm is sufficient. In fact, smooth block decomposition on DC/LDBSW dictionary provides, thanks to DC atoms, a uniform residual signal around zero since the first OMP iteration (see figure 1).

On the contrary, blocks with contours (i.e. labelled as 1) are handled as regions of interest where distortions have to be minimized in order to achieve a good synthesis quality. To do this, OMP algorithm has to iterate until the error between the original and the approximated signals is under a fixed reconstruction error. The simple usage of depth map quadratic error can lead to suboptimal results since it only measures coding artifacts and does not reflect the real impact of the latter on the final rendering quality. Therefore, we make use of the quadratic error in the synthesized frame and not in the depth map itself. We particularly use the distortion metric of Kim et al. [1] that takes into consideration camera parameters and proves the proportional relation between the quadratic error in the synthesized view and the absolute error in the depth map.

4.2. Experimental results and analysis

Since the main use of depth maps is in view synthesis operations, experimentations are concerned with the evaluation of views that can be synthesized from already compressed depth images. The following experimentations consist in coding left and right views from *Breakdancers*, *Ballet* and *Champagne* sequences. The decoded views are then used for view synthesis using View Synthesis Reference

Software (VSRS) [12] of Nagoya University. We note that from each test data sets, the first 16 frames were used.

To evaluate the DC/LDBSW dictionary interest for compression performances, we compare results obtained by the proposed compression method with DC/LDBSW dictionary to those obtained by the same method with DC/LDBS dictionary. We particularly choose DC/LDBS dictionary for comparison since it is the most competitive one to DC/LDBSW dictionary in terms of sparsity, as shown in figure 2. Performances of the proposed scheme with DC/LDBSW dictionary are also compared to 3D-HEVC, the ongoing 3D compression standard. We do compare our method to the 3D-HEVC standard since it is the latest reference for comparison that includes latest efforts of 3D research community being approved by MPEG. We typically make use of 3D-HEVC Test Model version 4.1 (3D-HTM 4.1) [13] for which temporal and inter-view predictions are disabled because our method does not involve them.

The performances of candidate methods are compared in terms of rate-PSNR tradeoff of synthesized views. Moreover, we make use of the new human visual system based metric, Structural SIMilarity plus (SSIMplus) [14]. We also propose the visual evaluation of areas zoomed from synthesized views.

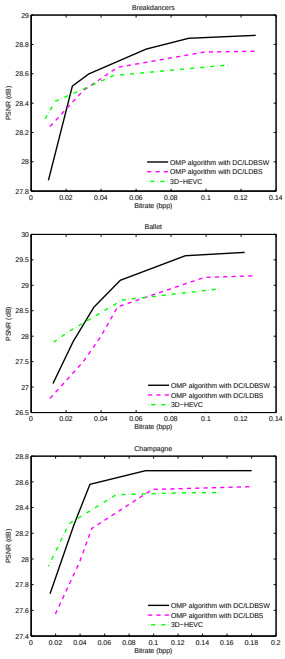


Figure 4. Rate/PSNR curves of *Breakdancers*, *Ballet* and *Champagne* synthesized views obtained from original textures and depth maps encoded using 3D-HEVC and the proposed method with DC/LDBS and DC/LDBSW dictionaries.

4.2.1 PSNR vs. Bitrate

Figure 4 depicts performances of candidate methods in terms of Bitrate-PSNR of synthesized views for *Breakdancers*, *Ballet* and *Champagne* sequences. Results of figure 2 have proved relevance of DC/LDBSW dictionary, against DC/LDBS one, in terms of sparsity. Figure 4 comes to show that DC/LDBSW dictionary is also better than DC/LDBS in terms of Bitrate-PSNR of synthesized views. Compared to 3D-HEVC, DC/LDBSW dictionary integrated within the proposed scheme provides better performances for medium and high bitrates, achieving a gain of $0.1\ dB$ at $0.1\ bpp$ for *Breakdancers*, $0.4\ dB$ at $0.08\ bpp$ for *Ballet* and $0.2\ dB$ at $0.1\ bpp$ for *Champagne*. However, 3D-HEVC allows better performances at very low bitrates since quantized values of wedgelet coefficients are restricted compared to indices of atoms dictionary that cannot be quantized.

4.2.2 SSIMplus Index

Since PSNR is a pure mathematical metric, we propose to use a new full-reference measure, SSIMplus. It provides real-time prediction of the perceptual quality of a video based on human visual system behaviors, video content characteristics, e.g. spatial and temporal complexity and video resolution, display device properties, e.g. screen size, resolution, and brightness, and viewing conditions, e.g. viewing distance and angle. Compared to most popular and widely used quality assessment measures, SSIMplus has shown a higher perceptual quality prediction accuracy and closer performances to Mean Opinion Scores [14].

Table 1 shows SSIMplus results of candidate methods obtained for test sequences at $0.01\ bpp$, $0.05\ bpp$ and $0.1\ bpp$. The evaluation is performed at these different bitrates that correspond to three critical values, namely low, medium and high bitrates. As already mentioned, the first 16 frames were used from each test data sets. It is clear from table 1 that the proposed compression scheme with DC/LDBSW dictionary produces better SSIMplus results against DC/LDBS dictionary. Confronted to the ongoing 3D-HEVC standard, the DC/LDBSW dictionary achieves a mean gain of 2 at $0.05\ bpp$ and 4 at $0.1\ bpp$. At $0.01\ bpp$, better results are performed by 3D-HEVC, achieving a mean gain of 2.

4.2.3 Zoomed areas

Besides PSNR and SSIMplus Human Visual System-based measure, figure 5 allows visual evaluation of areas zoomed from synthesized views of *Breakdancers*, *Ballet* and *Champagne* sequences. Since 3D-HEVC performances are better than those of our method at low bitrates, the visual evaluation is performed at $0.01bpp$. Compared to DC/LDBS dictionary, the proposed method with DC/LDBSW dictio-

Table 1. SSIMplus values of *Breakdancers*, *Ballet* and *Champagne* synthesized views obtained from original textures and depth maps encoded using 3D-HEVC and the proposed method with DC/LDBS and DC/LDBSW dictionaries at 0.01 bpp, 0.05 bpp and 0.1 bpp.

Sequence	Method	0.01 *bpp*	0.05 *bpp*	0.1 *bpp*
Breakdancers	3D-HEVC	29	34	38
	DC/LDBS	28	35	38
	DC/LDBSW	27	37	41
Ballet	3D-HEVC	32	42	43
	DC/LDBS	29	40	46
	DC/LDBSW	30	43	47
Champagne	3D-HEVC	39	46	47
	DC/LDBS	37	45	48
	DC/LDBSW	37	47	50
Mean	3D-HEVC	33	40	42
	DC/LDBS	31	40	44
	DC/LDBSW	31	42	46

nary can clearly achieve better visual synthesis quality with much less harmful distortions. Compared to 3D-HEVC, the proposed method with DC/LDBSW dictionary can achieve a competitive synthesis quality despite the outperformance of the latter at this bitrate in figure 4 and table 1. As examples, we distinguish areas circled in red where 3D-HEVC outperforms our method. The latter allows however better quality than 3D-HEVC for areas marked in green.

5. CONCLUSION

In this paper, we have combined depth maps compression and sparse representations that proved to be particularly relevant for compression purposes. Typically, we aimed to propose a redundant mixed dictionary adapted to depth maps sparse representation. Experimental results lead to the conclusion that it is the combination of a discrete cosine dictionary with well-localized linear B-spline wavelet atoms that yields a significant improvement in the sparsity of high-quality approximations of depth maps. Applied for depth maps compression, DC/LDBSW dictionary also shows good tradeoffs between bitrate and distortion of synthesized views. As perspective, we aim to propose an approach allowing a joint compression of the two components of MVD, namely texture and depth. In addition to the sparsity ratio, we aim to study the DC/LDBSW dictionary efficiency in terms of other comparison criteria that take into account the redundancy and the coherence of the proposed dictionary. Studies and comparisons to learned dictionaries are also in our scope.

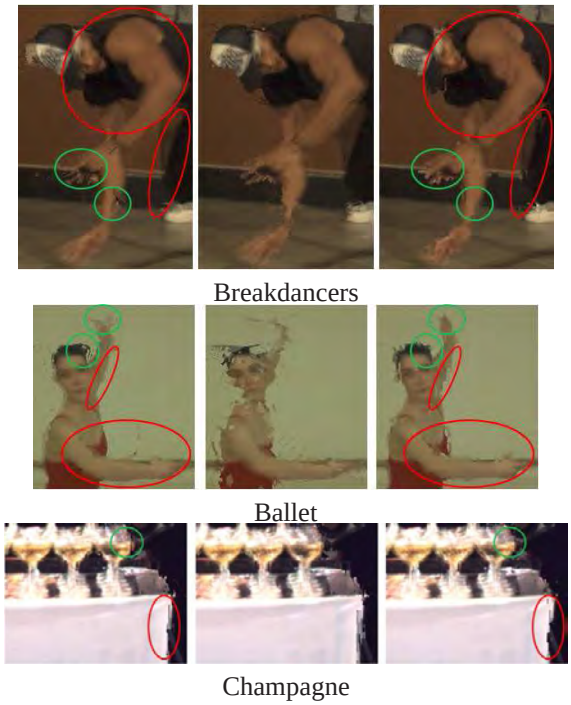


Figure 5. Zoomed areas of views synthesized from depth maps encoded at $0.01bpp$ using : 3D-HEVC (left), the proposed compression scheme with DC/LDBS dictionary (middle) and the proposed compression scheme with DC/LDBSW dictionary (right).

6. REFERENCES

[1] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *IEEE International Conference on Image Processing*, 2009.

[2] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. H. N. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Processing: Image Communication Journal*, 2008.

[3] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, H. F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing, Special Issue on 3D Video Representation, Compression and Rendering*, vol. 22, pp. 3366–3378, 2013.

[4] M. Maitre and M. N. Do, "Depth and depth-color coding using shape-adaptive wavelets," *Journal of Visual Communication and Image Representation*, vol. 21, pp. 513–522, 2010.

[5] G. Shen, W.-S. Kim, S. K. Naran, A. Ortega, L. Jaejoon, and W. HoCheon, "Edge-adaptive transforms for efficient depth map coding," in *Proceedings of IEEE Picture Coding Symposium*, 2010.

[6] G. Cheung, A. Kubota, and A. Ortega, "Sparse representation of depth maps for efficient transform coding," in *Proceedings of the 28th Picture Coding Symposium*, 2010.

[7] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform domain sparsification of depth maps using iterative quadratic programming," in *Proceedings of the IEEE International Conference on Image Processing*, 2011.

[8] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit recursive function approximation with

applications to wavelet decomposition," in *Proceedings of IEEE Asilomar Conference on Signals Systems and Computers*, 1993.

[9] *http://research.microsoft.com/en-us/downloads/*.

[10] R. M. F. i Ventura, P. Vandergheynst, and P. Frossard, "Low rate and flexible image coding with redundant representations," *IEEE Transactions on Image Processing*, vol. 15, pp. 726–739, 2006.

[11] K. Klimaszewski, K. Wegner, and M. Domanski, "Influence of views and depth compression onto quality of synthesized views," *ISO/IEC JTC1/SC29/WG11, M16758*, UK, 2009.

[12] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," *ISO/IEC JTC1/SC29/WG11 MPEG2008/M15377*, France, 2008.

[13] *https://hevc.hhi.fraunhofer.de/svn/svn 3DVCSoftware/tags/HTM-4.1*.

[14] A. Rehman, K. Zeng, and Z. Wang, "Display device-adapted video quality-of-experience assessment," in *Electronic Imaging, Human Vision and Electronic Imaging XX*, 2015.

Low Latency Rendering in Augmented Reality Based on Head Movement

Jung-Bum Kim Joon-Hyun Choi Sang-Jun Ahn Chan-Min Park

Samsung Electronics
34 seong-chon, 06765, Seoul, Republic of Korea
{jb83.kim, jh53.choi, sjun.ahn, chanmin.park}@ samsung.com

ABSTRACT

In Augmented Reality, AR, the latency is a huge problem that disrupts immersive experience especially when head-worn devices are involved. Rendering of virtual objects generally accounts for major proportion of the latency in AR. In this paper, we identify the problem caused by the latency and observe human perception of virtual objects during head movement to find opportunities to reduce the latency. We propose solutions that reduce the latency of rendering by introducing Level of Detail (LoD) concept based on head movement. Experimental results show our approach is effective to decrease the latency of rendering.

Keywords

Augmented Reality, low latency rendering, Level of Detail, head movement.

1. INTRODUCTION

Augmented Reality, AR, has significantly emerged in recent years. The advent of head-worn devices including [Ocu12a, Goo12a, Sam15a] has demonstrated usefulness of AR technology in various areas such as education, game, and location-based service [Van10a]. Thanks to the immersive experience that head-worn devices provide, AR has been recently gaining enormous attention [Zho08a]. In this paper, we examine properties that prevent users from getting absorbed in AR when head-worn devices are engaged. Head-worn devices, in AR perspective, are very different from stationery devices and hand-held devices, because users change their view direction very quickly. Rapid change in head direction is one of the major sources of common problems in AR. Since AR requires a variety of operations including object recognition, object tracking, and rendering, it inherently yields some latency. In contrast, the latency of perceiving real world is nearly zero, which indicates there could be a gap between real world and virtual objects. We define this gap as *realism gap* which describes inconsistency of virtual objects overlaying onto real world. We concentrate on rendering to reduce the realism gap by analyzing human perception of virtual objects upon rapid head movement. Our observation implies that AR users with head-worn devices are vulnerable to notice degradation of quality of virtual objects when they rapidly rotate their head. In order to reduce the computation amount required for rendering, we apply Level of Detail concept [Heo00a] in accordance with head movement. That is, we intentionally decrease the quality of virtual objects to the level which AR users are not able to detect. In addition, virtual objects closer to eyes are more influenced by rapid head rotation. Therefore, in our approach, virtual objects closer to eyes are subject to more degradation of quality. To validate our approach, we establish a test platform that consists of a smartphone with a camera, various sensors, and a Gear VR which transforms a smartphone to a head-worn device. Experimental result shows that our approach is effective to reduce the latency of rendering during head movement.

2. RELATED WORK

The latency problem in AR systems has been identified by Donald T. Azuma *et al.* in [Azu97a]. The paper analyzed common problems caused by the latency in AR systems. Zhou *et al.* [Zho08a] have explored researches in a recent decade regarding major techniques in AR including tracking, interaction, and display.

In Computer Graphics, many researchers have proposed approaches to reduce the latency of rendering. Level of Detail [Heo00a] is a common solution for the purpose of rendering optimization. Various works including [Xia97a, Lin96a] have presented LoD based approaches for reducing the rendering latency. In addition to LoD, there have been various optimization techniques for rendering including real time occlusion culling analyzed by S. Coorg et al. [Coo97a] and a shader simplification technique proposed by P. Sitthi-amorn *et al.* [Sit11a].

More revolutionized head-worn devices with an optical see-through display such as Microsoft HoloLens [Hol16a] have been recently released and are expected to be the mainstream.

However, general rendering optimization approaches do not take into account AR and head worn devices.

3. PROBLEM OF LATENCY IN AR

Most AR systems recognize real objects and augment relevant information on the real objects by rendering virtual objects in various forms including images, texts, and 3D models. A virtual object has a relative location to a particular real object to provide specific information about it. A label on top of a real object can be an example of a virtual object that has a relative location [Azu03a]. For immersive experience, it is important to consistently preserve a relative location between a virtual object and a real object. However, operations in AR such as object recognition, and rendering of virtual objects inherently require certain amount of the latency to accomplish its purpose. On the other hand, the latency of perceiving a real object is nearly zero. Figure 1 illustrates the difference of the latency between real world and virtual objects.

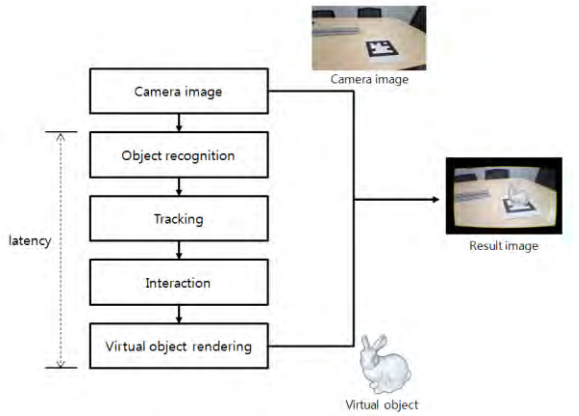


Figure 1. Latency difference between a camera image and a virtual object.

There is insignificant latency to acquire an image of real world. In order for a virtual object to be rendered, a large amount of the latency for various operations including object recognition and virtual object rendering is inevitable.

It is very difficult to retain assigned relative locations of virtual objects to real objects because of the difference of the latency. In other words, there is a spatial gap between virtual objects and real objects. We define this gap as *realism gap* which describes inconsistency of virtual objects overlaying onto real world. It gets worse when a head-worn device such as a smart glass and a head mount display is involved, because users change their view direction by head movement. Suppose that a user makes a rapid head turn. What a user sees in real world obviously changes in accordance with the transition. However, due to the latency, a virtual object still remains at a location where it was until rendering of a virtual object finishes. This realism gap not only disturbs immersive experience but also causes motion sickness. Reducing the latency to alleviate the realism gap is one of the keys for successful implementation of AR.

4. REDUCING LATENCY OF RENDERING

In order to reduce the latency, we concentrate on rendering of virtual objects. We observe how a user perceives a virtual object in rapid head movement. The first observation is that human visual system is vulnerable to notice insignificant change in a virtual object in the middle of rapid head movement. In fact, researches including [Bri75a] in various areas such as Vision, and Neurosciences have already explored a phenomenon called saccadic suppression which refers inability to detect changes in objects during rapid change in view direction. Our first observation complies with the researches. The second observation is that it is more difficult to notice change in a closer virtual object while a user rapidly rotates a head. From the observations, we conclude that it's acceptable to control quality of virtual objects in the middle of head movement to reduce the latency for rendering so that human visual system is unable to detect change in quality.

In Computer Graphics, Level of Detail, LoD, is one of the well-known approaches to reduce the amount of computation by controlling quality of virtual objects. Generally, the way LoD applies is based on distance of virtual objects from a camera [Lin96a]. LoD is able to reduce the amount of computation for rendering by decreasing quality of virtual objects distant from a camera. In this paper, to reduce the latency for rendering of virtual objects in AR, we apply Level of Detail particularly based on head movement in two aspects: 1) the amount of angular velocity of head rotation, 2) distance of virtual objects from a head.

The first criterion, angular speed of head rotation, makes sense according to numerous researches about saccadic suppression. Angular speed of head rotation determines quality of virtual objects by the following exponential function:

$$y = -b^{x-d} + 1 \qquad (1)$$

x is angular speed of head rotation. We define angular speed of head rotation as the amount of change in angle in degree in a second. y represents quality of virtual objects. Values of quality of virtual objects are normalized. And value 1 indicates the original quality. b plays a role to determine a curvature of the function. d is a constant that denotes the maximum angular speed of head rotation. Maximum angular speed of head rotation is obviously limited because a human's ability to move a head is constrained. In case of head-worn devices, angular speed of head rotation practically ranges from 0 to 90 degree per second. We experimentally conclude that exponential functions are appropriate to account for the human visual perception in the middle of head rotation. Experiments we perform imply that decline of the ability to perceive change in virtual objects accelerates, as angular speed of head rotation increases.

The second criterion, distance of a virtual object from a head, also helps reduce quality of virtual objects. A projection that transforms virtual objects is perspective, which means 3D space distorts in such a way that closer areas are larger than farther areas after transformation. Therefore, it is more difficult to recognize movement of closer objects in the middle of head rotation. This approach that further decreases quality of closer objects is the opposite of general application of LoD. In general, farther objects have lower quality because they are less noticeable. However, in case of head-worn devices, we decrease quality of closer objects more, because rapid head rotation has more influence on closer objects. The function that determines quality of virtual objects from distance of virtual objects is as follow:

$$y = -\frac{1}{ax} + \frac{1}{f-n} + 1 \qquad (2)$$

x is distance of a virtual object as an input. f denotes the farthest distance from a head. n is the nearest distance from a head. A perspective projection in 3D rendering typically accompanies a farthest plane and a nearest plane. a is a coefficient that determines a curvature of the function.

The final quality of a virtual object is defined as multiplication of outputs from (1) and (2). Therefore, quality of a virtual object is finally defined as

$$y = (-b^{x_1-d} + 1)(-\frac{1}{ax_2} + \frac{1}{f-n} + 1) \qquad (3)$$

x_1 is angular speed of head rotation and x_2 is distance of a virtual object. The rest of variables and constants are already explained above.

For practical application, for head rotations with angular speed less than a particular value, it is possible for users to notice change of quality. Therefore, quality of virtual objects retains at angular speed of head rotation less than a threshold. We plan to experimentally identify an optimal threshold value.

To degrade quality of virtual objects, our approach introduces mesh simplification technique from [Tur92a]. The technique shrinks a mesh by merging a particular number of vertices into one vertex.

Although our approach does not involve the general way of applying LoD that decreases quality of farther objects, combined with the general way of applying LoD, more reduction of computation is possible.

5. EXPERIMENTS

To validate our approach, we establish a test platform that consists of a smartphone and a Gear VR which transforms a smartphone to a head-worn device. This headset is an affordable type of AR devices, as smartphones typically contain various sensors, a high resolution display, and a camera. Figure 2 shows the device included in the test platform. The test platform also has software implementation of AR including object recognition, object tracking, and rendering.

Figure 2. A head-worn device used for experiments

The detail of device specification is as follows. CPU is a combination of Cortex-A53 quad-core 1.5 GHz and Cortex-A57 quad-core 2.1 GHz. GPU is Mali-T760MP8 which includes 8 cores. The dataset used contains a set of virtual objects in a form of 3D meshes. The 3D meshes consist of 3,352,500 vertices. As head-worn devices require stereo images, our implementation renders two images for left and right eyes respectively. The resolution of each image is 1024x1024. Virtual objects in the dataset are lit by one directional light.

We measure the amount of time that rendering takes as the latency, in order to evaluate the improvement

of our approach. Experiments also involve a conventional approach which renders virtual objects with constant quality for comparison. In experiments, constants b, d, a, f, and n in the equation (3) are set to 1.05, 90, 0.1, 2000, and 0 respectively.

The experimental result in figure 3 represents the relationship between angular speed of head rotation and the latency. We also alter distance values to observe the effect of distance to the latency. Distance values used are 15, 30, and 60. The latency of the conventional approach remains constant regardless of angular speed of head rotation or distance of a virtual object. The proposed approach significantly reduces the latency of rendering, as angular speed of head rotation increases. The latency decline accelerates to approximately 60 degree/sec, than decelerates. When the distance value is 60, the distance of a virtual object does not affect the latency. However, the latency at 10 degree/sec drops from 222 msec to 142 msec, when angular speed of head rotation increases to 55 degree/sec. Distance of a virtual object contributes to reduction of the latency as well. When angular speed of head rotation is 60 degree/sec, the latency at distance value 60 is roughly 22% of the latency at distance value 15.

Figure 5 illustrates output images of the proposed approach. A virtual object, Stanford Bunny, is rendered on top of a real object, a marker. It is possible to observe that a virtual object with the original quality in Figure 5 (a) changes to a version of lower quality in Figure 5 (b) during head rotation. We observe that it's hard to notice the quality degradation during head rotation.

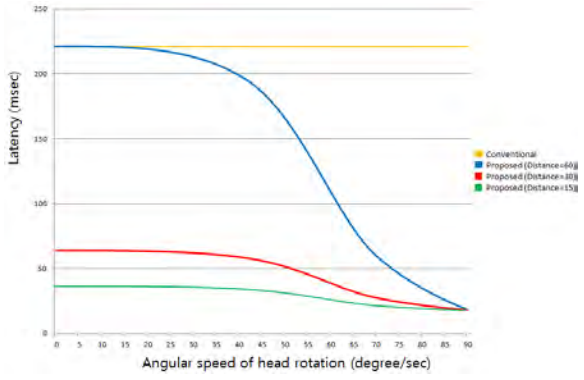
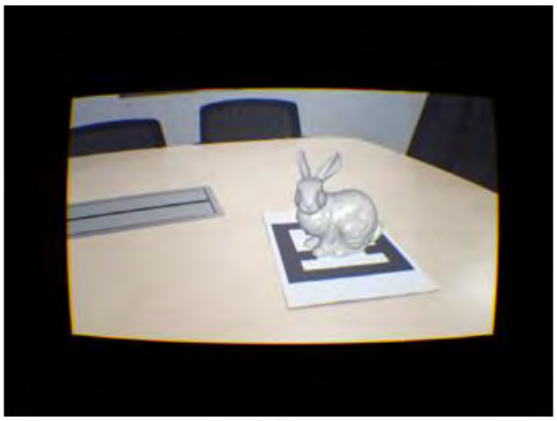


Figure 3. The latency of rendering from angular speed of head rotation and distance of a virtual object.



(a)

(b)

Figure 4. Quality change of a virtual object before (a) and during (b) head rotation.

6. Conclusion

The latency is one of critical problems in Augmented Reality. In this paper, we concentrate on reducing the latency of rendering, because rendering of virtual objects accounts for significant proportion of the entire latency in AR. By applying LoD based on head movement, the proposed approach successfully reduces the latency of rendering of virtual objects to 8% at best case.

7. Future work

We plan to continue developing and extending our approach for practical application. Since rendering of virtual objects is more complicated in practical cases, more decent control of quality is essential for efficient rendering of virtual objects. Our approach is expected to apply LoD to more factors such as display resolution, texture quality, ray tracing and so forth, in addition to the complexity of meshes. With more sophisticated control of quality, we plan to use real and practical datasets to present the effectiveness of the proposed approach.

In addition, to enhance other components of our implementation such as object recognition and object tracking, we consider employing a better solution such as Vuforia [Vuf12a].

8. REFERENCES

[Ocu12a] Oculus Rift, https://www.oculus.com, 2012.

[Goo14a] Google Cardboard Viewer, https://www.google.com/get/cardboard/, 2014.

[Sam15a] Samsung Gear VR, http://www.samsung .com/global/galaxy/wearables/gear- vr/, 2015.

[Van10a] Van Krevelen, D. W. F., & Poelman, R. A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality, 9(2), pp. 1-20, 2010.

[Zho08a] Zhou, Feng, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, pp. 193-202, 2008.

[Heo00a] Heok, Tan Kim, and Daut Daman. A review on level of detail. Computer Graphics, Imaging and Visualization, CGIV 2004. Proceedings. International Conference on. IEEE, pp. 70-75, 2004.

[Azu97a] Azuma, Ronald T. . *A Survey of Augmented Reality*. Teleoperators and Virtual Environments 6.4, pp. 355-385, 1997.

[Xia97a] Xia, Julie C., Jihad El-Sana, and Amitabh Varshney. Adaptive real-time level-of-detail based rendering for polygonal models. Visualization and Computer Graphics, IEEE Transactions on 3.2, pp. 171-183, 1997.

[Lin96a] Lindstrom, Peter, et al. Real-time, continuous level of detail rendering of height fields. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, pp. 109-118, 1996.

[Coo97a] Coorg, Satyan, and Seth Teller. Real-time occlusion culling for models with large occluders. Proceedings of the 1997 symposium on Interactive 3D graphics. ACM, pp. 83-ff, 1997.

[Sit11a] Sitthi-Amorn, Pitchaya, et al. Genetic programming for shader simplification. ACM Transactions on Graphics (TOG) 30.6, Article No.152, 2011.

[Hol16a] Microsoft HoloLens, https://www.microsoft .com/microsoft-hololens/en-us, 2016.

[Azu03a] Azuma, Ronald, and Chris Furmanski. Evaluating label placement for augmented reality view management. Proceedings of the 2nd IEEE/ACM international Symposium on Mixed and Augmented Reality. IEEE Computer Society, pp. 66, 2003.

[Bri75a] Bridgeman, Bruce, Derek Hendry, and Lawrence Stark. Failure to detect displacement of the visual world during saccadic eye movements. Vision research 15(6), pp. 719-722. 1975.

[Lin96a] Lindstrom, Peter, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust, and Gregory A. Turner. Real-time, continuous level of detail rendering of height fields. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, pp. 109-118, 1996.

[Tur92a] Turk, Greg, and Marc Levoy. Zippered polygon meshes from range images. Proceedings of the 21st annual conference on Computer graphics and interactive techniques. ACM, 1994.

[Vuf12a] Vuforia AR, http://www.vuforia.com, 2012.

An efficient 3-D environment scanning method

Kin Hong Wong Ho Chuen Kam Ying Kin Yu* Sheung Lai Lo
Department of Computer Science and Engineering
The Chinese University of Hong Kong
*Hong Kong
{khwong, hckam}@cse.cuhk.edu.hk, {ykyu.hk, lester2345}@gmail.com

Kwan Pang Tsui
Department of Mechanical and Automation Engieering
The Chinese University of Hong Kong
Hong Kong
warrentsui@outlook.com

Hing Tuen Yau
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong
billyauhk@gmail.com

ABSTRACT

In this paper, we discuss an idea of a system that can capture the 3-D model of a large area using only one single Kinect 3-D range sensor plus a stationary master camera. In operation, the Kinect is placed at different key positions to capture the local 3-D models, while a stationary master camera is situated behind the Kinect to find the current pose of the Kinect range sensor. Traditionally, a large scene can be scanned by moving the Kinect sensor across the whole area. Then the models obtained can be combined using motion capturing and pattern matching methods. However, the accuracy deteriorates when the area is too large or the environment does not provide enough features for registration. In our proposal, we place the Kinect at different key positions to obtain a number of local models. A dual-face checkerboard is placed on the top of the Kinect sensor in a way that the pattern can be seen from both the front and rear sides but not blocking the view of the Kinect. The pose of the board and the Kinect is estimated by a pose estimation algorithm using the images captured by the master camera. Since the embedded RGB-camera in the Kinect cannot see the checkerboard, a method based on a mirror is devised to determine the relative pose between the board and the embedded RGB-camera. Finally, we can combine all the 3-D local models and the pose information obtained to build up the complete global model. Various parts of the idea have been tested. We plan to integrate all parts and build a complete system for building the 3D map of a shopping mall or a museum in the future.

Keywords

Rotation averaging; mirrors; camera calibration; virtual reality development

1 INTRODUCTION

Obtaining the 3-D model of a small area can be achieved by a low cost 3-D scanner such as the Kinect camera. There is also a huge demand on the 3-D digitization of larger environment for virtual reality or 3-D navigation applications. Currently, a popular method is to scan the scene by moving the sensor manually for a distance to obtain the model by the software called Kinfu [Pir11]. However, the known problem of this approach is that the result may deteriorate if the scanning region is too large. Moreover, in order to achieve a reasonably good result, the scanning process is best to be performed by experienced technicians who are able to handle the scanner steadily for a long time. This creates a problem in deployment and execution. In this paper, we report a simple but yet effective method to solve the problem.

The setup is illustrated in Figure 1. In the proposed system, we employ only one Kinect sensor. It is to be placed at different key positions at different times to cover a large target area. One extra static camera, called the master camera in the world coordinate system, is needed to determine the current pose of the Kinect sensor. Pose estimation is achieved by placing a dual-face checkerboard on the top of the Kinect sensor. A standard pose estimation method [CW05] is used to obtain the position and orientation of the

checkerboard with respect to the master camera.
Even the poses of the Kinect sensor can be determined by the method explained above, however, there is an extra problem. Since the checkerboard and the Kinect cannot be aligned at the same position, we need to determine this small pose difference. This gives a certain difficulty to the overall system. Firstly, the checkerboard has to be observable from behind the Kinect by the master camera. It cannot be placed in front of the Kinect to block its view. The solution is that we fix the checkerboard on top of the Kinect so it can be seen on both (front and rear) sides. The board has two faces and its images are the same on both sides. Hence it is called the dual-face checkerboard. Using this scheme, we need to determine the pose between the dual-face board and the Kinect. Traditional extrinsic parameter calibration methods cannot be employed because the Kinect camera cannot observe the checkerboard directly. To tackle the task under this special arrangement, we propose to make use of a mirror together with the method by [LKL+15] to solve the problem. The procedure is discussed in the Section 3.
With the equipment setup mentioned above, the 3-D reconstruction process can be carried out as follows: Firstly, we need to choose a number of key positions. So by combining the models obtained at these key positions, the whole area of the target 3-D space can be covered. Then, at each key position, the Kinect sensor is operated to obtain the local 3-D model accurately by some existing approaches [Pir11] hence the localized 3-D structure can be calculated. Since a checkerboard is attached to the top of the Kinect sensor, the master camera is used to obtain the 3-D pose of the Kinect in the master (or world) coordinate system. Finally, we can combine all these local results into the world coordinate frame and we get the global 3-D model of the large environment.
The major contributions of this work are:

- The setup is low cost; only a single Kinect sensor and a normal digital camera are needed.
- The result is accurate compared to the traditional handheld scanning method, especially if the area is large or lacking features in some areas.

Our paper is organized as follows. The background of the research is discussed in section 2. The theories used in this work are discussed in section 3. The experimental result is shown in section 4. Section 5 concludes the work.

2 BACKGROUND

2.1 Structure From Motion

Structure from Motion (SfM) is an important problem in the field of computer vision. It has been studied

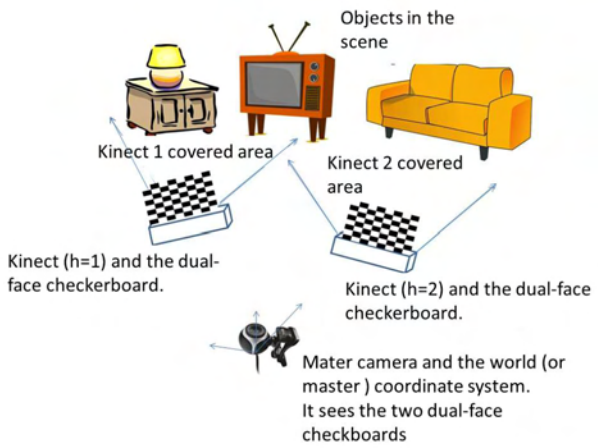


Figure 1: Overall setup - master camera and kinects

for more than a decade, for examples in the literature [JAP99], [Nis00], [HZ03], [FLP04], [YWC05], [YWC06] and [SSS08] etc.. The target is to find the 3-D model of an object from 2-D pictures. SfM is still an ongoing research topic. Most techniques rely on the correspondences between 3-D points of the object and its 2-D projection on the images. In the process, both the camera pose and the 3-D structure are computed. In order to find the correspondences among the input images, interest points are first located and extracted by the feature detectors like [HS88] and [Low04]. Features such as points, corners, edges and even planes can be tracked and extracted. Based on the features, the 3-D strcutrue can be estimated using bundle adjustment, for example [CW05].

2.2 Reconstruction by Kinect

Nowadays, there are lots of range sensors available in the market. They are suitable for performing computer vision tasks such as 3-D reconstruction. The Microsoft Kinect [Zha12] is a popular consumer grade range sensor for games and digital entertainment. Since it is inexpensive and its precision is high enough, it has been widely adopted by researchers in the field [CPF+12], [SHBS11], [TJRF13]. The underlying technology of a 3-D scanner is the use of Laser or Infra-red beam. The output is a point cloud consisting of the depth information that represents the structure of the target object. Triangles, polygon planes or curvature structures, can be constructured based on the obtained points through the process of surface reconstruction. One of the most straight-forward method is Triangle Strip devised by Zhang et. al. [ZZC+13]. It tackles the task by filling up the gaps along the two neighboring rows of horizontal pixels with multiple consecutive triangle plane surfaces. Triangle Strip offers an efficient way to reconstruct the object surface but the output is of relatively low quality. Another

algorithm having a better performance is Poisson Surface Reconstruction [KBH06]. It inserts hundreds of extra pixels in between the direct neighboring points. The insertion of pixel points is based on the original neighboring point curvature and tangential level to build up smooth object surfaces. In this way, it can produce high quality 3-D models. The main disadvantage is its long compuatation time, thus not suitable for real-time processing. Besides, there are other traditional algorithms to reconstruct 3-D surfaces, such as Ball Pivoting [BMR+99] and Power Crust [ACK01].

2.3 Long sequence reconstruction

There are some studies on the 3-D reconstruction of a large indoor environment. Kinect Fusion [NIH+11] is one of the most popular system for real-time surface mapping and tracking [IKH+11]. Depth information generated from the Kinect is used for pre-processing. After getting surface vertices and normal maps, a pose estimation algorithm is executed to calculate the camera pose in the scene. With Iterative Closest Point (ICP) procedure [BM92], a 6-DOF motion of the sensor is found by aligning the points in the current frame with respect to the previous ones. There is a major limitation of the Kinect Fusion system. As it relies on the ICP procedure for point matching, the model scanning process fails if it is applied to a plane with few features or a shiny surface [CKN+14].

3 THEORY

3.1 Over view of the system

The idea of our approach is illustrated in Figure 2. First we need to calibrate the pose between the dual-face checkerboard and the Kinect. It cannot be achieved by the usual camera calibration methods because the checkboard is not observable from the Kinect. To recover the pose between the Kinect and the checkerboard, we can employ a mirror-based technique similar to the one described in [LKL+15]. After this procedure, the rotation and translation between the dual-face checkerboard and the Kinect sensor can be found. Then we can use our setup to find the complete 3-D model of the environment. Since the dual-face checkerboard is very thin and the patterns on both sides are the same, the model and the image of the dual-side checkerboard are the same no matter which side you are looking at it. The procedure for 3-D reconstruction of the environment is described as follows. We first place the Kinect at a key position $h = 0$. The camera obtains the image of the dual-face checkerboard at the back of the Kinect. Using the pose estimation algorithm, the pose parameters (R_h, T_h) can be obtained. At the same time, the local 3-D model $M^{(h)}$ is also captured by the Kinect. We repeat the process for all h. After all models $M^{(h)}$ are obtained, we can put them into the coordinate system of the master camera (or world coordinate frame). In the following sessions, we will describe the details of (1) the pose estimation method between the master camera and the dual-face checkerboard and (2) a mirror-based pose determination approach to find the pose of the dual-face checkerboard relative to the Kinect.

3.2 Pose estimation between the camera and the dual-face checkerboard

Pose estimation is to determine the pose (R = 3×3 rotation matrix and T = 3×1 translation vector) of a rigid body when the 3-D model feature points $M_{i=1,2,...I}$ are given, where I is 3 or above and each 3-D feature M_i is a 3×1 vector. The method is discussed in [CW05] and is a well-known approach using the Gauss-Newton least-squares scheme.

Input: Models $M_{i=1,2,...N}$ with N 3-D feature points at rotation R=I_3, translation $T = [0,0,0]^T$; N 2-D image points $x_{i=1,2,...N}$ of the 3-D features at time t
Output: Pose = $\theta = [\phi_x, \phi_y, \phi_z, T_x, T_y, T_z]^T$ where $[\phi_x, \phi_y, \phi_z]$ = rotation angles, and $[T_x, T_y, T_z]$ = Translations
$x_i = p(M_i, \theta)$, where $p(M_i, \theta)$ is the projection of model M_i of feature i with pose θ to image x_i
Loop until error is small or too many times

1: Initialse $\theta_{k=0}$ and find $p(M, \theta_{k=0})$
2: **for** $(k = 1; k < K; k++)$ **do**
3: Find image error $e_i \leftarrow \|p(M_i, \theta_k) - p(M_i, \theta_{k-1})\|$
4: $E_k \leftarrow [e_1, e_2, ..., e_N]^T$
5: and Jacobian $J \leftarrow \frac{\partial E_k}{\partial \theta}$
6: $\Delta\theta_k \leftarrow J^{-1} * E_k$
7: Break if $\Delta\theta$ is small enough
8: $\theta_{k+1} \leftarrow \theta_k + \Delta\theta_k$
9: **end for**
10: Return θ_k

3.3 Pose computation between the dual-face checkerboard and the Kinect

Since the dual-face checkerboard is required to determine the pose of the Kinect with respect to the master camera, there is a need to find out the relative pose (rotation=R_b, translation=t_b) between the dual-face checkerboard and the Kinect because they cannot be aligned perfectly. The idea is illustrated in Figure 3. This can be achieved using a mirror and the method is shown in Figure 4. Users are required to perform this procedure just once since only one Kinect sensor is used in the scanning operation. There are two steps in this procedure.

3.3.1 Step 1: Pose estimation through a mirror

In this step, the Kinect and the checkerboard are stationary. The user places the mirror at n different positions. At each mirror position, the user takes a picture of the dual-face checkerboard through the mirror using the Kinect-RGB-camera. After this procedure, we have n pictures for pose estimation. For examples, the camera calibration toolbox [Bou11] or the pose estimation algorithm [CW05] can be applied to determine the pose of the checkerboard relative to the master camera. Then $R_{i=1,2,..n}$ rotations are obtained. However, please be noted that the rotations $R_{i=1,2,..n}$ obtained through capturing the checkerboard through the mirror are needed to be converted back to the corresponding improper rotations $\tilde{R}_{i=1,2,..n}$ using the formulas (equation 5) found in [LKL+15]. For example, an easy test to see if a rotation is improper or not is to see if $det(R) = -1$.

3.3.2 Step 2: Weiszfeld algorithm

Assuming the rotation between the Kinect and the dual-face checkerboard is R_b, the Weiszfeld algorithm [HAT11] is able to find this from $\tilde{R}_{i=1,2,..n}$ obtained in the above step. The procedure is shown in Algorithm 1.

Algorithm 1 Weiszfeld algorithm to find R_b

Input: $\tilde{R}_{i=1,2,..,N}, R_{init}$
Output: R_b
$R_{b(k=0)} = R_{b(init)}$
Define: $V = eig\_vector = V, D = eigen\_value$

1: **repeat**
2: $[V,D] \leftarrow eig(R^T_{b(k)}\tilde{R}_i)$
3: Select index c such that $D(c) ==$ -1
4: $n_i \leftarrow V(:,c)$
5: $w \leftarrow log(R^T_{b(k)}\tilde{R}_i(I - 2n_i n_i^T))$
6: $\delta \leftarrow \frac{\sum_{i=1}^{i=n} \frac{w}{\|w\|}}{\sum_{i=1}^{i=n}(1/\|w\|)}$
7: $R_{b(k+1)} \leftarrow exp(\delta)R_{b(k)}$
8: **until** diff. between $R_{b(k+1)}$ and $R_{b(k)}$ is very small
9: **return** $R_b \leftarrow R_{b(k)}$

The above algorithm is based on the method proposed by [LKL+15] and [HTDL13]. It is a rotation averaging scheme to find the optimal rotation. In the algorithm, the inputs are the rotations collected from the mirror images of the checkerboard. Rotation averaging has been studied in [HAT11], [LKL+15], [HTDL13], [KIFP08], [Huy09], [CG13] and [HTDL13]. Since the board is not directly observed by the camera but only its reflected image, the rotation obtained $\tilde{R}_i$ is required to be transformed back to the normal view by the formulation $R_b^T\tilde{R}_i(I - 2n_i n_i^T)$. It is computed by steps 2 to 5

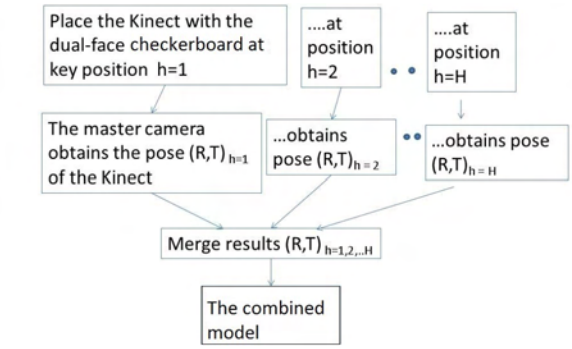


Figure 2: Overview of the approach

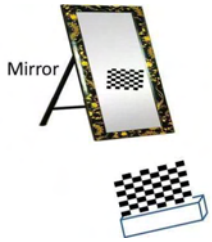


Figure 3: The Kinect camera can see the checkerboard through the mirror

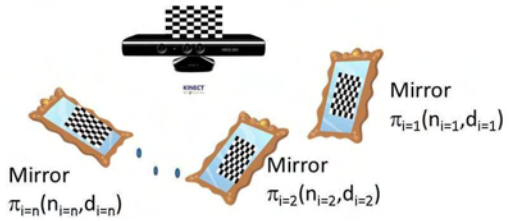


Figure 4: Calibration of the dual-face checkerboard and camera by repositioning the mirror

of Algorithm 1 and the method is described in [LKL+15] . Since the relative rotation of the pose R_b between the board and the camera is fixed and unchanged, the only change is the mirror orientation which will affect $\tilde{R}_i$. If we have enough samples of $\tilde{R}_i$, we can find R_b using an iterative scheme. So we need to identify a metric for evaluating the similarity between two rotations. A recent formulation is to use the metric proposed by Huynh [Huy09]. Together with the Weisfield algorithm, this method [LKL+15] can find the pose between the board and the camera efficiently and accurately. After R_b of the pose is found, the translation t_b can also be found by a simple linear formula using Equation(13) of [LKL+15].

4 EXPERIMENTS

4.1 Simulation for rotation averaging

The rotation averaging algorithm is used to find the rotation component of the pose between the dual-face checkerboard and the Kinect camera. We have carried out a simulation test to evaluate the performance of the rotation averaging method. The test was implemented in MATLAB 7.11 on a desktop computer.

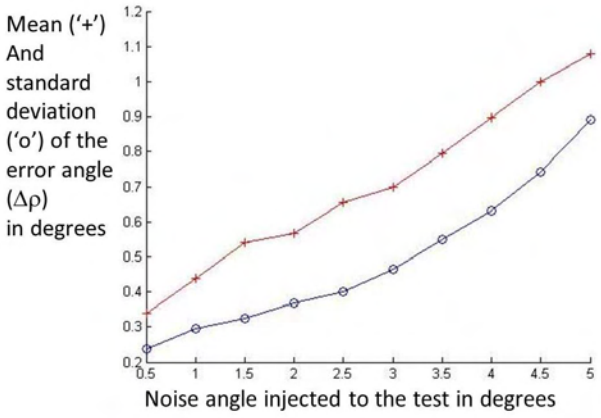


Figure 5: Simulation test result: The red +_line is mean error angle $\Delta\rho$ in degrees. The blue o_line is standard deviation of error angle $\Delta\rho$ in degrees.

In each test, we created a certain ground-truth rotation $R_{b\_ground\_truth}$ of the pose between the dual-face checkerboard and the Kinect-camera. This rotation was used to create 15 mirrored rotations $\tilde{R}_i$ based on the mirror reflection formula in Equation 4 in [LKL+15]. Noise in terms of rotation angles was injected into each of the rotation axis of input rotation $\tilde{R}_i$ with a standard deviation of from 0.5 to 5 degrees. Then we used the rotation averaging algorithm in Algorithm 1 to find the rotation matrix R_b. 1000 tests were carried out for each level of noise injected. To show and analyze the performance of the system, we plot the mean and standard deviation of the error angle $\Delta\rho$ against noise injected in Figure 5 .The error angle $\Delta\rho$ is the angle of the axis-angle representation of ΔR_{error}, where $\Delta R_{error} = R^T_{b\_found} * R_{b\_ground\_truth}$. The mean and standard deviation of the results are shown in Figure 5. As we can see from Figure 5, the error angle $\Delta\rho$ is still small even under noisy conditions. It shows that the rotation averaging method can compute the rotation of the pose accurately.

4.2 Mirror simulation toolbox

To let the readers further investigate into the mathematical properties of a mirror and the process of virtual object creation, we have developed a mirror simulation toolbox based on the formulas in [RBN10]. In the simulation test, we can create 3-D model points and a planar mirror in arbitrary positions. Then the corresponding virtual points and images can be formed and displayed. The 2-D virtual image points are captured by the real camera (in the simulation) and are then passed to a pose estimation algorithm [CW05], [Bou00] to find the pose of the virtual object with respect to the real camera. It is noted that the pose found should contain the improper rotation matrix since it is calculated from the virtual object. However, the camera does not know whether

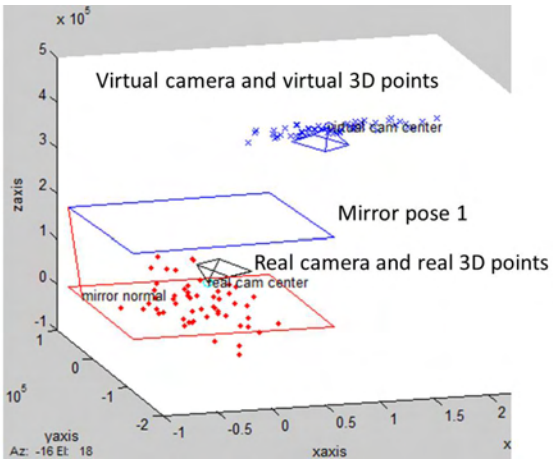


Figure 6: The First Sample Case of Our Mirror Simulation

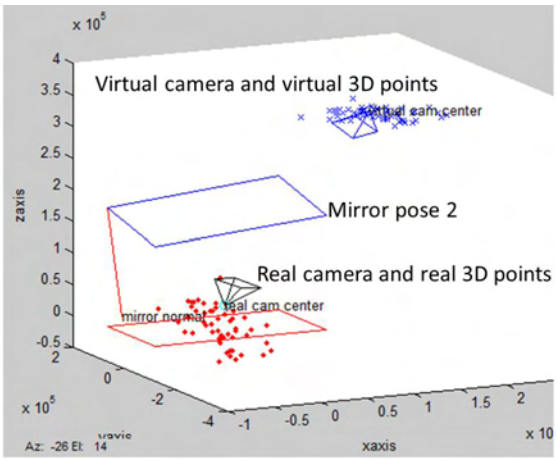


Figure 7: The Second Sample Case of Our Mirror Simulation

the object is a real one or from the mirror. So we need to convert the rotation (from proper to improper rotation) using the function described by Equation 5 in [LKL+15]. R_i is the converted rotation and is used in Algorithm 3.2. In this way, mirrors at different positions can be generated, resulting in a set of images of the virtual object. R_i can be calculated from these images using a pose estimation algorithm. If we can have enough R_i for $i \leq 3$, Algorithm 3.2 can be applied to find the rotation R_b between the real camera and the object. Screen shots of the toolbox are shown in Figures 6 and 7 . The toolbox visualizes the mirroring process. Files related to this research can be found at
https://appsrv.cse.cuhk.edu.hk/∼khwong/www2/conference/2016/WSCG2016/WSCG2016.html

4.3 Real image experiment

We tested the proposed idea shown in Figure 1 using two Kinects at two different key positions, which are about 60 cm apart. The pose between the two Kinects are then calibrated as described in Section 3.

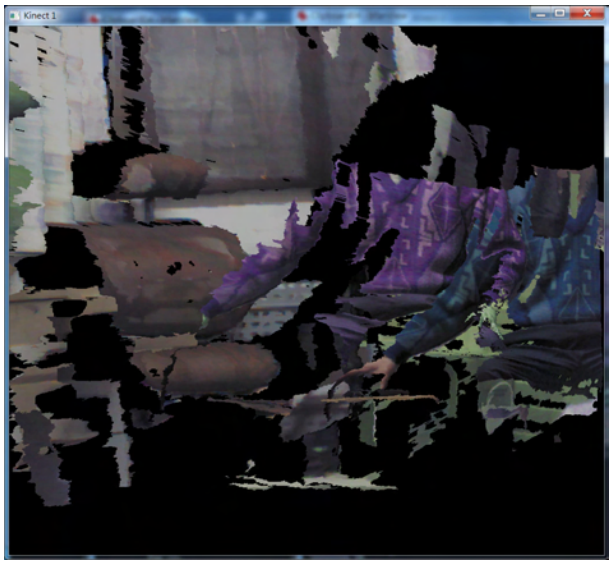


Figure 8: The 3-D point clouds before merging and they are separated

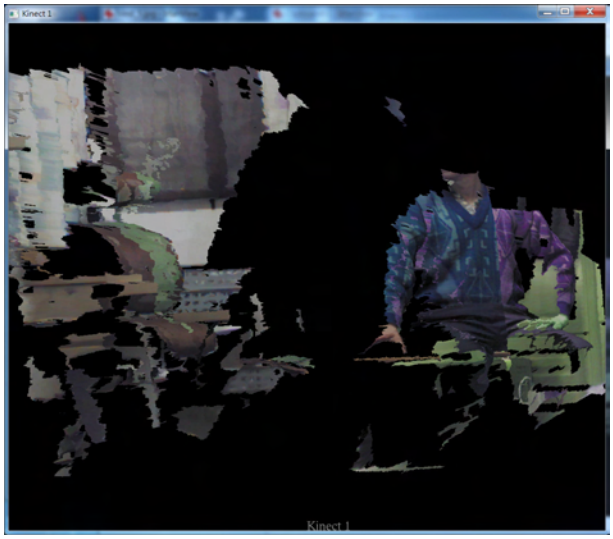


Figure 9: The 3-D point cloud combined using the computed pose parameters

The 3-D point clouds captured by the two Kinects are merged to generate a large 3-D model with the pose computed. The results before and after merging are shown in Figure 8 and Figure 9, respectively. It is demonstrated that the idea is to be feasible. Our next plan is to build a complete system to make it become a working system.

4.4 Comparison with an existing method

Our method is compared to an existing method called KinFu [Pir11] in terms of stability and usability. KinFu [Pir11] is a popular public domain software. It can be used to scan a large indoor area. The main problem of using Kinfu is that the operator is required to move the Kinect sensor very slowly. In our experience, the translation and rotation motion

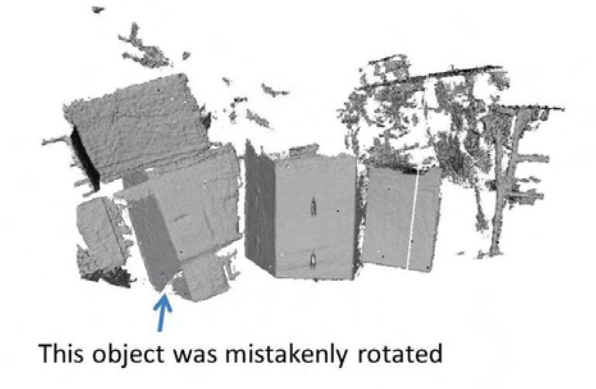


Figure 10: This is a case when Kinfu fails. An object in the scene was mistakenly rotated.

of the Kinect should be smaller than 0.5 meters or 10 degrees per second during operation, respectively. As our method requires the Kinect to be placed at a few stationary positions in the environment, it is convenient to use and the performance is relatively stable. Figure 10 shows the 3-D point cloud of a scene accquired by KinFU. It fails to capture the 3-D model of the environment correctly. It is because during scanning the sensor is rotated slightly for about 10 degrees, objects with vertical edges in the point cloud are mistakenly rotated. KinFu is unstable and not easy to be handled.

Method	**Cost**	**Easy to use**	**Accuracy**
Ours	Low, a common PC will do the job	Easy	High
KinFu	High, requires GPU	Not easy, needs the user to operate with care	High but can become inaccurate when handled incorrectly.

5 CONCLUSION

In this research, we have dicussed a system that can capture a large environment based on a two-level approach. At the first level, a Kinect sensor is placed at different positions of a large environment to obtain a number of local 3-D models. A dual-face checkerboard is attached to the Kinect so its pose relative to a stationary master camera can be estimated and recorded. Finally, all local models, each obtained by the same Kinect placed at different positions, are combined to become the complete wider view global

model. We have also adopted a method using mirrors and rotation averaging to calibrate the pose between a camera and an object that the camera cannot observe directly; the object can only be seen by the camera through a mirror. The pose information is computed using a rotation averaging algorithm called the Weisfield Algorithm [HAT11]. A toolbox of the mirror image formation process and rotation averaging algorithm is also developed to help us use the mirror-based techniques. Unlike existing approaches based on motion tracking methods, our system is easy to deploy, relatively stable and low cost. A test is carried out to show that we can combine local models to become a larger model. The system can be used in many applications such as virtual and augmented reality. Although it is not a complete system yet, we are confident that the idea is feasible and we will work on it to build a complete system in future. For example, we will apply the proposed techniques to the reconstruction of a large shopping mall or a museum.

REFERENCES

[ACK01] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2):127–153, 2001.

[BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[BMR+99] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359, 1999.

[Bou00] Jean-Yves Bouguet. Matlab camera calibration toolbox. *http:\ www. vision. caltech. edu\ bouguetj\ calib doc*, 2000.

[Bou11] Jean-Yves Bouguet. Camera calibration toolbox for matlab, h ttp. *www. vision, caltech, edu*, 2011.

[CG13] Avhishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 521–528. IEEE, 2013.

[CKN+14] S Chumplue, T Kondo, I Nilkhamhang, P Bunnun, and M Sato. 3d room reconstruction using visual odometry guided kinectfusion with rgb-d camera. 2014.

[CPF+12] Ross A Clark, Yong-Hao Pua, Karine Fortin, Callan Ritchie, Kate E Webster, Linda Denehy, and Adam L Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & posture*, 36(3):372–377, 2012.

[CW05] Michael Ming-Yuen Chang and Kin Hong Wong. Model reconstruction and pose acquisition using extended lowe's method. *Multimedia, IEEE Transactions on*, 7(2):253–260, 2005.

[FLP04] Olivier Faugeras, Quang-Tuan Luong, and Theo Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.

[HAT11] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3041–3048. IEEE, 2011.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. Citeseer, 1988.

[HTDL13] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[Huy09] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[JAP99] Tony Jebara, Ali Azarbayejani, and Alex Pentland. 3d structure from 2d motion. *Signal Processing Magazine, IEEE*, 16(3):66–84, 1999.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[KIFP08] Ram Krishan Kumar, Adrian Ilie, Jan-Michael Frahm, and Marc Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[LKL+15] Gucan Long, Laurent Kneip, Xin Li, Xiaohu Zhang, and Qifeng Yu. Simplified mirror-based camera pose computation via rotation averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1247–1255, 2015.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[Nis00] David Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Computer Vision-ECCV 2000*, pages 649–663. Springer, 2000.

[Pir11] Michele Pirovano. Kinfu–an open source implementation of kinect fusion+ case study: implementing a 3d scanner with pcl. *Project Assignment, 3D structure from visual motion, University of Milan*, 2011.

[RBN10] Rui Rodrigues, Joao P Barreto, and Urbano Nunes. Camera pose estimation using images of planar mirror reflections. In *Computer Vision–ECCV 2010*, pages 382–395. Springer, 2010.

[SHBS11] John Stowers, Michael Hayes, and Andrew Bainbridge-Smith. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE International Conference on*, pages 358–362. IEEE, 2011.

[SSS08] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[TJRF13] Yasuhiro Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5182–5189. IEEE, 2013.

[YWC05] Ying Kin Yu, Kin Hong Wong, and Michael Ming-Yuen Chang. Recursive three-dimensional model reconstruction based on kalman filtering. *IEEE Transactions on Systems, Man and Cybernetics: Part B (Cybernetics)*, 35(3):587–592, 2005.

[YWC06] Ying Kin Yu, Kin Hong Wong, and Michael Ming-Yuen Chang. Merging artificial objects with marker-less video sequences based on the interacting multiple model method. *IEEE Transactions on Multimedia*, 8(2):521–528, 2006.

[Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, 2012.

[ZZC+13] Zhou Zhang, Mingshao Zhang, Yizhe Chang, El-Sayed Aziz, Sven K Esche, and Constantin Chassapis. Real-time 3d model reconstruction and interaction using kinect for a game-based virtual laboratory. In *ASME 2013 International Mechanical Engineering Congress and Exposition*, pages V005T05A053–V005T05A053. American Society of Mechanical Engineers, 2013.

DEVELOPMENT OF COMPUTATIONAL PROCEDURE OF LOCAL IMAGE PROCESSING, BASED ON THE USAGE OF HIERARCHICAL REGRESSION

V.N. Kopenkov
Samara University
443068, Samara, Russia
vkop@geosamara.ru

ABSTRACT

The article deal with technology of digital images processing on the base of non-linear algorithms. This approach allows to construct the efficient procedure of local image processing. The aim of this research is to workout an algorithm of processing images with predetermined computational complexity and the best quality of processing on the existing data set avoiding a problem of retraining or lesstraining. To achieve this aim we use local discrete wavelet transformation and hierarchical regression to construct local image processing procedure on the base of a training dataset. Moreover, we workout method to estimate the necessity of finishing or continuing the training process. This method is based on of the functional of full cross-validation control which allow to construct processing procedure with predetermined computational complexity and veracity, and with the best quality.

Keywords

local processing, hierarchical regression, interval estimate, function of complete sliding quality control.

1. INTRODUCTION

The tasks of image processing and signal analysis need to be solved in different fields of human activity [Soi09, Woo05, Pra07]. Local processing of digital images is one of the most important kinds of transformation in the theory and practice of digital image processing and computer vision.

Historically, the first processing procedures used local linear methods that allow the construction of optimal (in some sense), processing procedures [Soi09, Pra07]. However, the taking into consideration of new digital signal processing tasks (processing of video, audio, satellite images, etc.), problems of processing large amounts of information (satellite images, remote sensing data, hyperspectral data, multi-dimensional signals), processing in real time, and needs of rising of processing efficiency resulted in the necessity of using nonlinear type of transformations [Soi09, Hai06]. One of the most common approach currently in use is the implementation of the cybernetic principle of the "black box" (the terms of other authors is the processing via recognition, processing basing on precedents and so on). In this case transformation itself and its parameters are determined by the analyzing of the input and output signals, or images.

The classic approach to construction of approximately universal procedures of local adaptive digital signal and images processing, which implements the principle of "black box" is based on usage of artificial neural networks technique [Hai06]. An alternative, but substantially less researched version of described task solution based on using of a hierarchical computational structure, such as the decisions tree and regressions tree. [Bre84, Kop12]. This paper develops the idea of the creation of universal mechanism for the construction of local non-linear computational processing procedures based on a hierarchical scheme and features based on local discrete wavelet decomposition of the image.

In addition, the methodology of decision-making on stopping the learning process, as well as the veracity of the obtained results are also presented in the article. Usually, for estimation of the generalization capability and selection a stop learning rule for processing procedure the Vapnik–Chervonenkis statistical theory is used [Vap74]. This theory interconnects the three parameters of training: training error, veracity (reliability), and the length of training dataset. But estimates of statistical theory are highly overestimated and ignore the potential rearrangement of training and testing dataset elements. A more efficient way to estimate the generalization capability is the use of Vorontsov combinatorial theory [Vor04], which is based on evaluation of the functional of full cross-validation, assuming verification all possible combinations of division dataset into training and testing parts. The

correct solution to the task of construction of an image processing procedures which takes into account all combination of sets of training and control data is unrealizable in practice because of the giant search of variants of different combinations.

The paper is organized as follows. The first section devoted to subject introduction. The task specification and description of proposed solution, as well as a scheme of processing process are presented in the second section. Description and structure of the algorithm of construction of local image processing procedure and its parameters is presented in the third section. The fourth section is devoted to the description of methodology which allows to determine the rule for stop the process of formation and busting various combinations of training and control samples and stop construction process in general. And finally, conclusions, recommendations, acknowledgements, and references are presented in the end of the paper.

2. TASK SPECIFICATION

2.1. Image local processing model

Model of the local image processing technology, which implement the principle of "black box" (processing through the recognition or based on precedents), suggests decomposition of the transformation in two stages: the formation of the image fragment description (local features computing) and calculation of transformation results. The general scheme of image processing is shown in Fig 1.

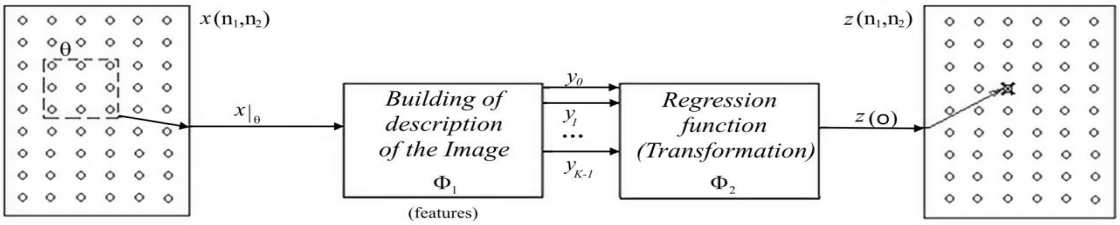


Figure 1. Local image processing scheme.

The main task in the first stage is the formation features (some specific set of image properties) for predetermined local image fragment – $\bar{y} = (y_0, y_1, ..., y_{K-1})^T$, $\bar{y} \in \mathbf{R}^K$ on the base of transformation $\Phi_1 : \mathbf{R}^{M_1 \times M_2} \to \mathbf{R}^K$.

These features are used to calculate the result of transformation $\Phi_2 : \mathbf{R}^K \to \mathbf{K}$ (and to generate the resulting image *Z*) during the second stage of processing.

The whole construction process is based on the processing precedents – a set of matched pairs of images $\left\{ x|_{\theta(n_1,n_2)}, z(n_1,n_2) \right\}_{(n_1,n_2):\theta(n_1,n_2)\subseteq\Theta}$ (which are usually called *training dataset*) in order to minimize the processing error:

$$\varepsilon = \frac{1}{|\Theta|} \sum_{(n_1,n_2)} \left\| z - \Phi_2(\Phi_1(x)) \right\| \to \min_{\Phi_1,\Phi_2}, \qquad (1)$$

where Θ – the image domain, $\theta(n_1,n_2) \subseteq \Theta$ – restriction to the local fragment size $M_1 \times M_2$: $\theta(n_1,n_2) = \left\{ (n_1+m_1, n_2+m_2) : m_1 = \overline{0, M_1 - 1}, m_2 = \overline{0, M_2 - 1} \right\}$.

2.2. Characteristics of the decision

The most known solution of described task is based on the usage of artificial neural networks. Such approach has some special features, advantages and disadvantages which are well described in detail [Hai06]. The alternative technology of the processing procedure construction is based on special hierarchical computational structures, such as *regression trees* and *decision trees* [Bre84, Kop12]. These trees are the hierarchical structures consisting of 2 types of vertices - non-terminal vertices which define a partition of features domain, and terminal vertices which store a regression function.

The procedure based on regression trees has some advantages in comparison with the neural networks:

- automatic correction of "architecture" of the transformation;
- automatic selection of local features which is result of the partition process;
- finitely of the building and tuning process (computational efficiency);
- ease of tuning of the regression parameters in the terminal vertex.

There are some restrictions for the practical implementation. At first, the most important task on the stage of image features calculation in a "sliding window" mode is the task of development of a computationally efficient algorithm for this calculation. Moreover, this algorithm should allow consistently increase the features number up to whole system, because the traditional algorithms of the defining of the effective features subset based on iterative methods and computationally inefficient. At second, the main task on the stage of designing of hierarchical regression is the development of an algorithm to automatic construction of processing procedures on the base of the training dataset which

be able to avoid the retraining and insufficient training problems.

2.3. Choosing of the features types

We used the family of signal characteristics on the base of local wavelet discrete transformations (DWT) of the signals and images as an image features set. Such features have the following characteristics:

- ✓ existence of the computationally efficient calculation algorithm [Kop08];
- ✓ complete description of the input signal;
- ✓ consistent obtaining and usage of features removes the problem of iterates on the features set.

Issues related to the features formation on the base of local DWT algorithms, as well as their advantages and specialty in relation to local image processing tasks, are considered in the work [Kop08].

The classic scheme for fast calculating of local wavelet transformation (FWT) is based on Mallat scheme [Hai06] and in accordance to the theory of a multiple-scale analysis can be represented as following equations:

$$w_{l+1}^{+}(p) = \sum_{n\in D_h} h(n-2p)\cdot w_l^{+}(n),$$
$$w_{l+1}^{-}(p) = \sum_{n\in D_g} g(n-2p)\cdot w_l^{+}(n),$$

where $p=\overline{1,N}$, N – length of the input signal, $h(n),\ g(n)$ – such filters that: $\sum_n h(n)=1$, $g(n)=(-1)^n h(-n+2t-1),\ (t\in Z)$, D_h, D_g – length of the filters, $l=\overline{0,\log_2 M}$ – wavelets levels, M – processing window size.

Concerning the image processing, the computational complexity of such algorithm for the wavelet levels $[L_1, L_2]$ can be evaluated [Kop08] as following:

FWT on the base of Mallat scheme: $U_1^{*}(L_1,L_2) = 8/3(2^{2L_2}-1)$;

Modified FWT: $U_2^{*}(L_1,L_2) \underset{N\to\infty}{\approx} 8L_2 - 5L_1 - 5$;

Recursive FWT: $U_3^{*}(L_1,L_2) = 13(L_2 - L_1 + 1)$.

3. ALGORITHM FOR CONSTRUCTION OF THE LOCAL IMAGE PROCESSING PROCEDURE

3.1. Regression tree construction

Technology of regression trees construction consists of the following stages:

1) Determination of parameters and features of hierarchical structure building process.

Necessary to select the vertices for further separation based on estimation error in them. Then determine the *parameters of vertices partitioning* (threshold and number of vertices to divide) and choose the "*best feature*" for partition. Such choice has to be done taking into account that the aggregation of "best feature" and parameters of the partition should provide maximum errors reduction.

In the case of linear regression:

$$f_j(\bar{y}) = \sum_{k=0}^{K-1} a_{jk} y_k + a_{jK}, \quad \left(y_k \in [y_{k,j}^{\min}, y_{k,j}^{\max}]\right), \qquad (2)$$

in the each terminal node, the processing error (1) can be estimated as:

$$\varepsilon_j(k,\alpha) = \min_{a^{*}} \left\| f_j^{*}(\bar{y}) - \bar{z}_j \right\|_{[y_{k,j}^{\min}, y_{k,j}^{\alpha}]} + \min_{a^{**}} \left\| f_j^{**}(\bar{y}) - \bar{z}_j \right\|_{[y_{k,j}^{\alpha}, y_{k,j}^{\max}]},$$

where $\bar{a}^{*}$, $f_j^{*}(\bar{y})$ – the regression coefficients and function (according to (2)) for the new left node;

$\bar{a}^{**}$, $f_j^{**}(\bar{y})$ – the regression coefficients and function (according to (2)) for the new right node;

α – the optimal threshold of partition;

k_j – the "best feature".

2) Calculation of the *regression coefficients* for each terminal vertex on the base of least squares method. We have to solve the system of linear algebraic equations by using all elements of the training dataset "fallen" into terminal vertex.

3) Checking the restrictions for computational complexity and estimation of processing quality on the base of test dataset.

3.2. Finishing of the construction process

To determine the stop parameters for algorithm construction process, we need to estimate the generalization capability of the local processing procedures.

There is a dataset: $\Omega = \{\bar{y}_j, z_j\},\ j=\overline{1,T}$:

$$\Omega^T = \Omega^s \cup \Omega^t, \quad s+t=T, \quad \Omega^s \cap \Omega^t = \varnothing. \qquad (3)$$

The quality of algorithm on a dataset:

$$\nu(a,\Omega) = \frac{1}{|\Omega|} \sum_{\omega_i\in\Omega} I(\omega_i, a(\omega_i)),$$

where a – regression algorithm, A – set of algorithms, μ – method (algorithm) of teaching based on a dataset: $\mu(\Omega) = a$,

$$I = \begin{cases} 1, & |a(\omega_i) - \Phi(\omega_i)| \ge \delta(\omega_i) \\ 0, & |a(\omega_i) - \Phi(\omega_i)| < \delta(\omega_i) \end{cases}.$$

Usually, for estimation of the generalization capability and limitations on the length of the training dataset the statistical theory is used [Vap74]. This theory is based on the analysis of functional of uniform deviation of the error rate in 2 samples:

$$P_{\varepsilon}^{st}(A) = P\left\{\sup_{a\in A}\left(\nu(a,X^t) - \nu(a,X^s)\right) > \varepsilon\right\}.$$

But, the statistical theory characterized by heavy reliance of estimates and, moreover, this theory not take into account the possible shuffles of the elements of the training and testing samples. A more efficient way to estimate of the generalization capability is the usage of combinatorial theory [Vor04] based on a functional of a full sliding control, which is invariant to arbitrary permutations of samples:

$$Q^{st}(\mu,\Omega^T)=\frac{1}{N}\sum_{n=0}^{N-1}\nu\left(\mu(\Omega_n^s),\Omega_n^t\right),\quad \left(N=C_T^s\right), \quad (4)$$

and takes into account all three factors: the characteristics of samples distribution, restored dependence and the teaching method.

3.3. Constructional process

Development of effective algorithm of the local image processing based on a hierarchical regression and such features as a local DWT of image requires simultaneous consideration of different performance indicators: the computational complexity of the procedure, its quality or processing error and generalization capability.

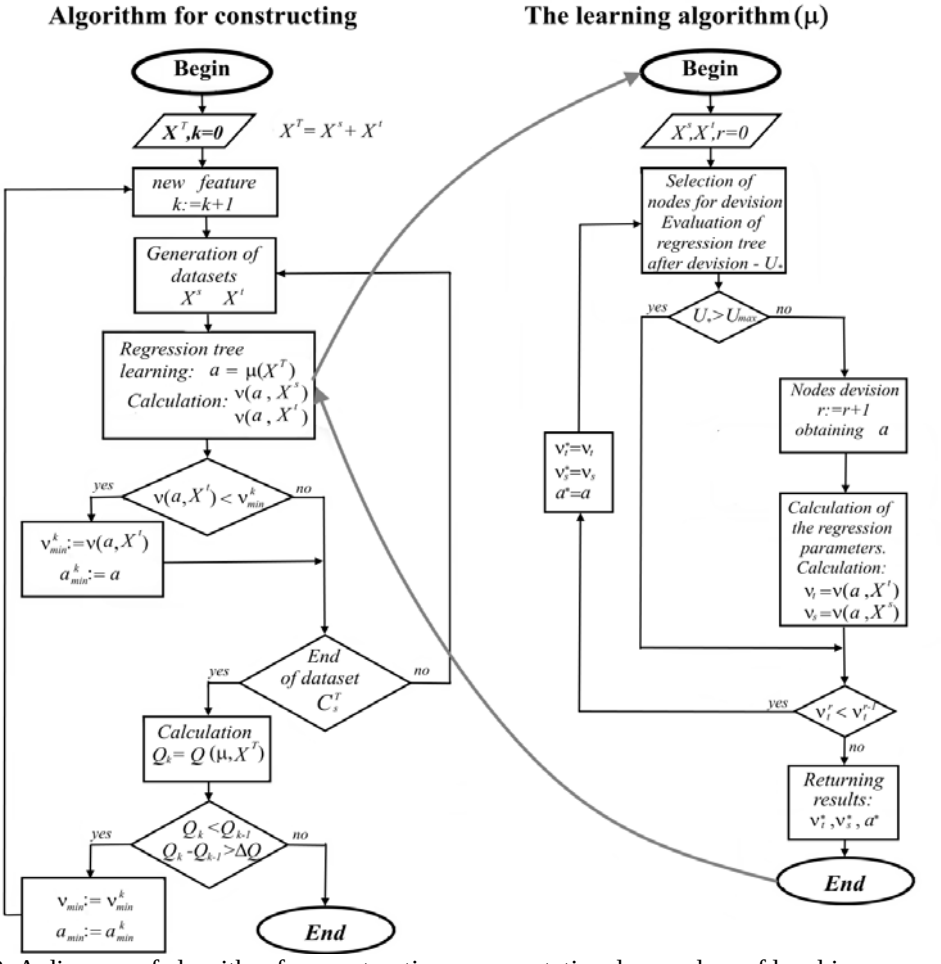


Figure 2. A diagram of algorithm for construction a computational procedure of local image processing.

The scheme of the algorithm of automatic construction of processing procedures is shown in Figure 2. The algorithm assumes the consistent accumulation of features as long as the functional of a full sliding control is decreasing (that means that a quality of processing is improving), and the computational complexity of the procedure remains in predetermined limits.

3.4. Experimental researches

As an experimental task, we consider the task of image filtration. The solution involves the use of local image processing procedures based on regression tree (RT) and artificial neural network (NN). The comparison of the processing quality and computational complexity of these algorithms is presented in table 1. As can be seen from the table, the proposed method of hierarchical regression has the better accuracy with essentially smaller computational complexity than well known neural network method.

NN	ε	10.92	10.78	10.69	10.67	10.63	10.62	10.62	10.64
	U	54	99	189	279	369	459	549	621
RT	ε	11.28	11.01	10.89	10.81	10.72	10.62	10.57	10.61
	U	31	38	41	44	46	48	50	52

Table 1. The comparison results.

4. METHODOLOGY OF STOPPING OF TRAINING PROCCESS.

4.1. Scheme of exhaustive search on datasets

Taking into account generalization capability of the local processing procedures based on a functional of a full sliding control [Vor04] we can estimate that the total number of all possible *N* decompositions of dataset is C_T^s. General scheme of construction procedure is shown in Figure 3.

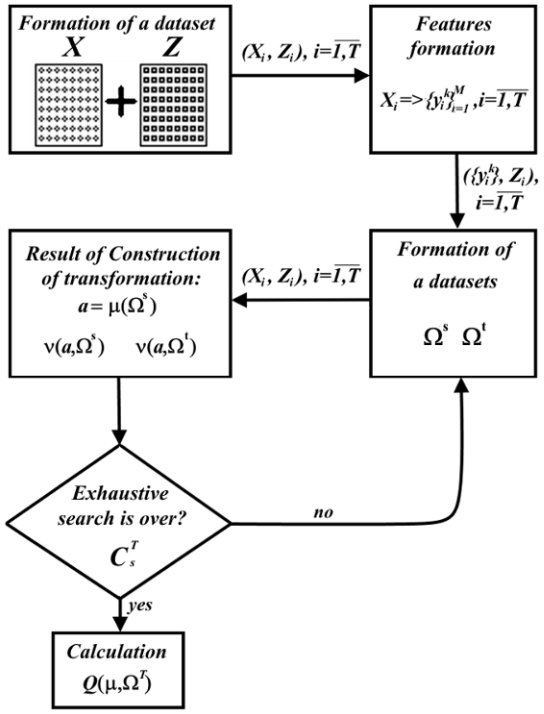


Figure 3. Schema of construction procedure with exhaustive search on datasets.

Is quite logical fact that in the case of images processing the construction of processing procedures which takes into account all combination of training and testing dataset is unrealizable because of the incredibly large busting on various combinations of datasets. Therefore, we have had to develop a method of determination rule for stop busting process on the base of a finite number of samples.

4.2. Scheme of exhaustive search on datasets

For sufficiently large sample volumes can be assumed that the error rate of the algorithm has a binomial distribution with *t* degrees of freedom (the length of the test dataset) and the probability of "success" = *p* (quality of the algorithm on a control set). In this way:

$$\nu(\mu(\Omega_n^s),\Omega_n^t) \sim Bin(t, p)\,.$$

Function of the probability is specified as:

$$p_v(r) = C_t^r\, p^r (1-p)^{t-r},\ \ r = \overline{0,t}.$$

Then the distribution of the functional full cross-validation is evaluated by:

$$Q^{st}(\mu(\Omega),\Omega) = \frac{1}{N}\sum_{n=0}^{N-1} \nu\left(\mu(\Omega_n^s),\Omega_n^t\right) \sim Bin(N\cdot t, p)\,.$$

Decision about continuing or stopping generation of different combinations training and control datasets and transition to the next subset of features can be taken based on the analysis of functional $Q_1^{st} \sim Bin(N_1\cdot t, p_1),\ \ Q_2^{st} \sim Bin(N_2\cdot t, p_2)$ for the different subsets of features. We decide whether to recalculate the feature space, or to stop the process of building a processing procedure under the assumption of $p_2 < p_1$, with veracity γ (and correspondingly $p_1 < p_2$, with the veracity $(1-\gamma)$).

In such case the quality of the algorithm on dataset Ω can be estimated as:

$$\nu(\mu(\Omega),\Omega^T) \sim \frac{1}{|\Omega|}\sum_{\omega_i\in\Omega} I(\omega_i,\mu(\omega_i)),$$

where $I(\omega_i,\mu(\omega_i)) = \begin{cases}1, & p\\ 0, & 1-p\end{cases}$.

Moreover, if $n >> 1$ (that is justified, because *n* is the number of objects and corresponds to the image size) and the λ is fixed, we obtain the Poisson distribution with parameter λ:

$$Bin(n, \lambda/n) \approx P(\lambda).$$

In this case to make a decision of stop generation process for various combinations of training and control datasets, and to transition to next features set we have to calculate confidence intervals for the expectation of a Poisson distribution for the functional full cross-validation on a datasets N_1 and N_2 in form:

$$\left[\lambda_1 - \frac{\tau_{1-\alpha/2}\sqrt{\lambda_1}}{\sqrt{N_1}}, \lambda_1 + \frac{\tau_{1-\alpha/2}\sqrt{\lambda_1}}{\sqrt{N_1}}\right]\left[\lambda_2 - \frac{\tau_{1-\alpha/2}\sqrt{\lambda_2}}{\sqrt{N_2}}, \lambda_2 + \frac{\tau_{1-\alpha/2}\sqrt{\lambda_2}}{\sqrt{N_2}}\right],$$

where $\tau_{1-\alpha/2}$ – quantile of distribution $N_{0,1}$ for level $1-\alpha/2$ $(\alpha = 1-\gamma)$.

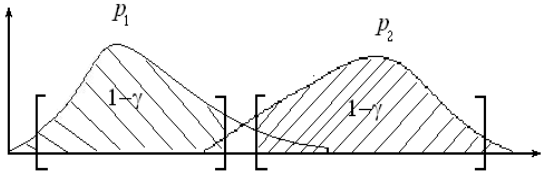


Figure 4. Calculation of confidence intervals.

The decision of stopping generation of different combinations training and control datasets and about the switching to the next subset of features is taken at

a moment when a separation of calculated confidence intervals on adjacent steps is achieved.

4.3. Illustration of the process of processing algorithm construction

Figure 5 shows an example of training of a regression tree, for different sets of features (**K**=1,2,3,...,12, with a gradual increase). The graphs show the noise reduction (ε^2/D_V) with the increasing of a regression tree depth (H_{av}). Figure 6 presents a statistics of process of regression tree construction on a various combinations of training and control datasets (group of points of each colors correspond to the optimum value of quality for a given set of features **K**=1,2,3,...,12, in the case of exhaustive search a some number of partitioning of a dataset Ω on training and control part (Ω_n^s, Ω_n^t), $n = 1, 2, ..., N$).

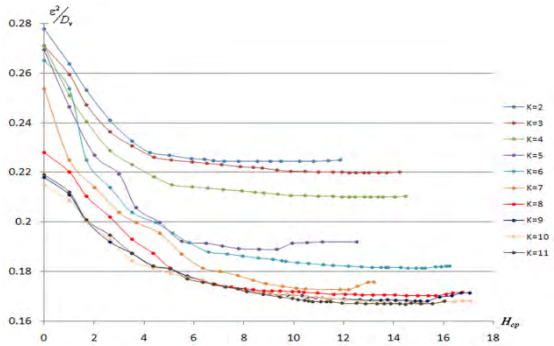


Figure 5. Process of training of processing procedure at various features space.

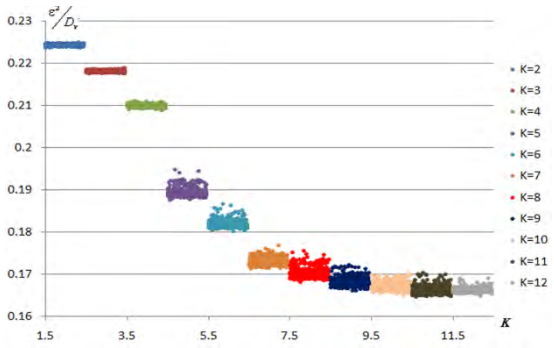


Figure 6. Statistics of quality of procedures for different combinations of training and control datasets.

Figure 7 shows a graph of the construction process of local image processing procedures, with confidence intervals, for the optimal values of quality, and Figure 8 – calculation of the required number of combinations of training/testing datasets for the making a decision of switching to the next set of features (the number of combinations required for the separation of the confidence intervals on the adjacent steps).

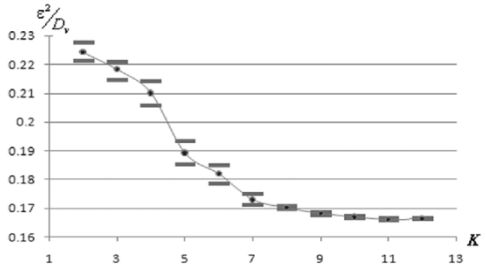


Figure 7. Construction of the processing procedure with confidence intervals.

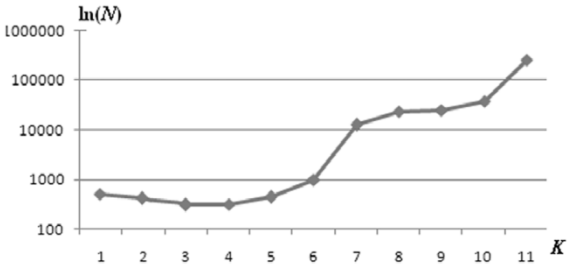


Figure 8. Calculating of combinations number.

5. CONCLUSIONS AND RESULTS

The paper presents an *efficient technology* that allows to realize automatic construction of computational procedure of local processing of digital signals/images. In accordance with the creation processes the constructed computational procedure has a specified complexity and the highest quality and the generalizing ability. The proposed method of estimation of the required number of algorithm training iterations and, as a consequence, the stopping rule of the formation different combinations of training and testing datasets based on their particular number allows to use the full functionality of combinatorial theory and a functional of full cross-validation control during the constructing (training) processing procedures, which are tuning on the bases of a training dataset. And as a result, possible to prevent problems of retraining/poorly trained processing algorithms and, at the same time, construct the local processing procedure with predetermined computational complexity and veracity, and with the best quality (for an existing training dataset).

6. ACKNOWLEDGEMENTS

This work was financially supported by the Russian Scientific Foundation (RSF), grant no. 14-31-00014 “Establishment of a Laboratory of Advanced Technology for Earth Remote Sensing”.

7. REFERENCES

[Soi09] Methods of computer image processing. Part II: Methods and algorithms / M.V. Gashnikov, N.I. Glumov, N.Yu. Ilyasova, V.V. Myasnikov, S.B. Popov, V.V. Sergeev, V.A. Soifer,

A.G. Hramov, A.V. Chernov, V.M. Chernov, M.A. Chicheva, V.A. Fursov. – Ed. by V.A. Soifer. – Moscow: "Fizmatlit" Publisher, 2009. – 784 p.

[Woo05] Woods, R. Digital Image Processing / R. Woods, R.Gonsales / - M: Technosphere, 2005. - 1072 p. (in russian)

[Pra07] Pratt, W. Digital image processing. Wiley, 4ed, 2007.

[Hai06] Haikin, S. Neural Networks: A Comprehensive Foundation / M.: «Vilyams», 2006. 1104 p.

[Bre84] Breiman L. Classification and regression trees / Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone.// Monterey, Calif., U.S.A.: Wadsworth, Inc. – 1984.

[Kop12] Kopenkov V.N., Myasnikov V.V. An algorithm for automatic construction of a local non-linear processing procedures images based on hierarchical regression. Computer optics, 36(2):257-266, 2012. (in russian).

[Vap74] Vapnik, V.N. The theory of pattern recognition / VN Vapnik, A. Chervonenkis / - Moscow: Nauka, 1974. (in russian)

[Vor04] Vorontsov K. A combinatorial approach to assessing the quality of training algorithms / / Mathematical problems of cybernetics / Ed O.B. Lupanov. Moskow. Fizmatlid. 2004. Vol. 13. p. 5–36. (in russian)

[Kop08] Kopenkov V. Efficient algorithms of local discrete wavelet transform with HAAR-like bases. Pattern Recognition and Image Analysis. Vol 18 No 4 2008 pp. 654-661.

[Kop14] Kopenkov V. On halting the process of hierarchical regression construction when implementing computational procedures for local image processing. Pattern Recognition and Image Analysis. Vol 24 No 4 2014 pp. 506–510

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Detection of Challenging Dialogue Stages Using Acoustic Signals and Biosignals

Olga Egorow
IIKT
Otto von Guericke University
39106 Magdeburg
Germany
olga.egorow@ovgu.de

Andreas Wendemuth
IIKT & CBBS
Otto von Guericke University
39106 Magdeburg
Germany
andreas.wendemuth@ovgu.de

ABSTRACT

Emotions play an important role in human-human interaction. But they are also expressed during human-computer interaction, and thus should be recognised and responded to in an appropriate way. Therefore, emotion recognition is an important feature that should be integrated in human-computer interaction. But the task of emotion recognition is not an easy one – in "in the wild" scenarios, the occurring emotions are rarely expressive and clear. Different emotions like joy and surprise often occur simultaneously or in a very reduced form. That is why, besides recognising categorial and clear emotions like joy and anger, it is also important to recognise more subtle affects. One example for such an affect that is crucial for human-computer interaction is trouble experienced by the human in case of unexpected dialogue course. Another point concerning this task is that the emotional status of a person is not necessarily revealed in his or her voice. But the same information is contained in the physiological reactions of the person, that are much harder to conceal, therefore representing the "true signal". That is why the physiological signals, or biosignals, should not be left unattended. In this paper we use the data from naturalistic human-computer dialogues containing challenging dialogue stages to show that it is possible to differentiate between troubled and untroubled dialogue in acoustic as well as in physiological signals. We achieve an unweighted average recall (UAR) of 64% using the acoustic signal, and an UAR of 88% using the biosignals.

Keywords

Emotion, affect, affective computing, emotion recognition, acoustic emotion recognition, biosignals

1 INTRODUCTION

One of the goals of human-computer spoken interaction is to become more and more similar to human-human interaction, turning the machine into an almost-human companion. For this matter, not only understanding what a human is saying is important, but also how it is said – the emotions of the human counterpart are an equally important part of interaction. This is why computers should be able to recognise and understand emotions. But this is a challenging task, since human emotions can range in a variety of dimensions. As a starting point, we can consider six basic emotions following the categorial model [EFE72]: happiness, surprise, fear, sadness, anger and disgust combined with contempt. But natural emotions comprise clearly more than that. Another classification of emotions, better suited for natural emotions, is the multidimensional scale of pleasure-arousal-dominance [Meh96]. But not all emotions are relevant for human-computer interaction. One emotion is especially important in this aspect: the feeling that humans experience when something unexpected happens during a dialogue. If an interaction suddenly becomes challenging for the human counterpart, it is crucial for the success of the dialogue to recognise it and to react in an appropriate way.

In this paper we show that even slight emotional changes occurring in naturalistic human-machine dialogue can be predicted from acoustic as well as from physiological signals. For this purpose, we use data of human-computer interaction obtained during a naturalistic multimodal Wizard-of-Oz (WOZ) experiment, consisting of unchallenging and challenging dialogue stages. This design allows us to look into problems that naturally arise from challenging human-machine interaction. For this purpose, we simplify the dialogue stages to two main classes: the *baseline* class containing normal interaction and the *trouble* class containing challenging interaction. We show that it is possible to automatically

discriminate these dialogue stages based on acoustic data as well as on biosignals.

The remainder of the paper is structured in the following manner: first, we give an overview on existing related work in section 2; in section 3 we introduce the data set used for our experiments; in section 4 we describe the feature extraction routines for both data types, and the classification process; in section 5 we present and discuss the achieved results; in section 6 we summarise the work and show some possibilities for future development.

2 RELATED WORK

Automatic emotion recognition on acoustic data has achieved considerable results in the last years. On acted data, like the standard Berlin Database of Emotional Speech [BPR+05], recognition rates of 85% are possible [SVE+09]. But in realistic tasks, emotions often cannot be divided in discrete categories of joy, sadness, anger, etc. The automatic classification of naturalistic data is difficult and achieves less impressive outcomes: for "in the wild" scenarios using audio only recognition results of about 33% UAR [RAE+14] to about 24% UAR [SVE+09] and 20% accuracy [SDS+13] for seven-class-problems are achieved. But even for human annotators emotion recognition from speech alone is not an easy task – for acoustic data without context, an average accuracy of 60% is achieved by human raters [Sch81]. To solve this problem, humans rely on other modalities of interaction besides speech, like facial expressions, gestures, etc. This applies also to automatic emotion recognition. For example, adding the dimension of co-speech gestures improves the results by 2 percentage points compared to speech alone [BBJ14]. Another experiment shows that using only facial expressions outperforms using only acoustics by almost 15 percentage points in terms of accuracy, and using both data types simultaneously in a bimodal system leads to a further improvement of 4 percentage points [BDY+04].

Additionally, there are physiological reactions related to emotions, like a racing heart, gasping and sweating palms. The links between biosignals such as heart rate, temperature, skin resistance and emotions have been known for a long time [ELF83] [LEF90] [CBL+00]. One big advantage of using physiological data is that biosignals can be obtained at all times – acoustic signals can only be obtained when the user is speaking. Another point is that physiological reactions, in contrast to reactions in voice, are much harder to conceal. That is why it is not surprising, that recognition of emotions on physiological signals achieves better results than only on acoustic data. This has been proven many times, for example for categorial emotions, where the recognition rate ranges from 49% to 75% for induced emotions [RBND06] [KBK04] [CW05], for the multidimensional valence-arousal scale with a recognition rate of over 90% [HGSW04], also for induced emotions, and for other scales like fun levels, with a recognition rate of 70% [YH08].

There are also multimodal approaches combining both physiological and acoustic data. One example is similar to our setting: a WOZ quiz show with four stages, corresponding to four classes on the valence-arousal scale (high and low for valence and arousal, respectively), where six different biosignals and acoustic feedback are used for emotion recognition [Kim07]. The results vary depending on the evaluation method (92%-69% accuracy for subject-dependent vs. 55% accuracy for subject-independent evaluation), but combining the physiological and acoustic features leads to an improvement in all cases. Although this direction seems promising, the research on this topic is still rare. One possible problem is that the gathering of naturalistic multimodal data is not a simple process, in terms of recording (e.g. problems concerning the synchronisation of data streams) as well as in terms of processing (e.g. fusion of data streams).

Most of the studies presented above deal with emotion recognition in general and investigate elicited emotions. But on the important topic of recognising trouble in human-machine communication, especially naturalistic communication, not much research has been done so far. The groundbreaking example is detecting trouble in acted and elicited interaction using acoustic data and detailed annotations containing part-of-speech (POS) tagging, dialogue acts, repetitions and syntactic boundaries, achieving 73% to 96% recall for different scenarios [BFH+03]. One of the scenarios described in this approach is a naturalistic WOZ experiment, here the data was separated into two classes: one class containing prosodic peculiarities and one class containing no prosodic peculiarities. This setup resembles the setup of our study, but in our case we rely on much simpler annotations of the acoustic data and, more importantly, on biosignals.

3 THE DATA SET

3.1 The LAST MINUTE Corpus

The LAST MINUTE Corpus [FMR+12][RFF+12][PRS+14] contains naturalistic multimodal recordings of German speaking subjects in a WOZ experiment. The setup of the experiment revolves around the preparations for an imaginary journey to an unknown place "Waiuku". Each experiment lasts about 30 minutes and consists of several dialogue stages, each triggered by a major event. A summary of the different dialogue stages can be seen in Table 1. First, the subjects are asked to introduce themselves to the "machine" in a "warm-up"

dialogue stage. After that, they are requested to imagine winning a summer trip to an unknown destination called Waiuku, and they have to pack a suitcase by choosing items from a list in a "listing" dialogue stage. There is also a time constraint: the trip begins immediately, and the subjects have only fifteen minutes for the packing process. After several minutes of packing there is another major event: the subjects learn that the suitcase has a weight limit, so they have to remove some of the packed items. This corresponds to the "challenge" dialogue stage. After that, the next major event occurs when the real destination of the trip is revealed: Waiuku lies in the southern hemisphere. Since the subjects now know that the trip is a winter trip instead of a summer trip, they have to re-organise their suitcase again. This dialogue stage is called the "Waiuku" stage. At the end of the experiment, there is a short "conclusion" stage. It is expected that the subjects experience different emotions during the dialogue stages, and also express them. It should be noted that like in any naturalistic scenario, the subjects may react differently, with reactions ranging from very expressive to very subtle.

Dialogue Stage	Trigger	Troubled?
Warm-up	Introduction request	No
Listing	Winning the trip	No
Challenge	Weight Constraint	Yes
Waiuku	Revealing destination	Yes
Conclusion	Concluding remarks	No

Table 1: Overview of the dialogue stages

The acoustic data is recorded using 2 directional microphones at 44100 Hz and stored in the wav format. The biosignal data is recorded using the NeXus-32 system[1]. From the various biosignals available from this system, it proved sufficient for our analysis to use electromyogram (EMG), skin conductivity (SC) and respiration (RSP). These biosignals could be obtained in sustained quality throughout the experiment.

3.2 Selecting the Data

From all the recordings of the LAST MINUTE Corpus we selected a subset containing the recordings of 19 subjects, of whom both the acoustic and the physiological data exist. The age and sex distribution of the subjects is nearly balanced, as shown in Table 2. The acoustic data set and the biosignal data set are divided into three subsets to enable subject-independent evaluation. The subsets for the acoustic data contain the same subjects as the subsets of the biosignal data, leading to a training subset containing data of 11 subjects, a development subset containing data of 4 subjects and a test subset containing data of 4 subjects each. These subsets are also nearly balanced regarding the distribution of age and sex, cf. Table 2. In the classification process, the models are built on the training subsets, fine-tuning the parameters of the classifier takes place on the development subsets in order to avoid overfitting to the test data, the test subsets are used to obtain the final classification results.

	Train	Dev	Test	Overall
Sex				
Female	5	2	2	9
Male	6	2	2	10
Age				
Young (< 30)	7	2	2	11
Elder (> 60)	4	2	2	8

Table 2: Distribution of sex and age of the subjects.

3.3 Dividing the Data into Classes

The hypothesis of this paper is that it is possible to automatically detect the different dialogue stages described above in both acoustic and biosignal data recorded during the experiments. For this purpose, we divide the data into two classes: the *baseline* class, denoting untroubled interaction in the warm-up, listing and conclusion dialogue stages, and the *trouble* class, denoting the challenge and Waiuku dialogue stages, where the subjects are expected to experience trouble during the interaction. It should be noted that no perception tests were conducted, therefore the data is not annotated regarding the level of trouble expressed by the subject. The labels consist only of the dialogue stages. Therefore, the trouble class contains different levels of trouble. We will present the different levels using two examples from the challenge dialogue stage.

The first example shows two snippets from a dialogue, here the subject is an elderly woman. In the first part, she is – for the first time – informed by the Wizard that the weight limit is reached:

Wizard: *A swimsuit or bikini cannot be added, otherwise the maximum weight limit prescribed by the airline would be exceeded. Before other items can be selected, you must provide enough space in your suitcase. For this, already packed items can be unpacked. On demand, you can get a list of the already selected items.*

Subject: *Yes, uh ((pause)) I would like ((pause)) take out a pair of shoes.*

Wizard: *Your statement cannot be processed.*

[1] http://www.mindmedia.info/CMS2014/products/systems/nexus-32

After she fails to remove some items, she selects some more items and gets the same message from the Wizard again. Now she seems frustrated:

Wizard: *Before other items can be selected, you must provide enough space in your suitcase. For this, already packed items can be unpacked. On demand, you can get a list of the already selected items.*

Subject: *hm ((moaning)) yes (.) then I want to hear the chosen items again please, I told you I want to unpack shoes.*

The second example shows the dialogue of a young woman, who also learns that there is a weight limit:

Wizard: *Before other items can be selected, you must provide enough space in your suitcase. For this, already packed items can be unpacked. On demand, you can get a list of the already selected items.*

Subject: *Remove inflatable boat.*

Wizard: *One inflatable boat was removed, you can continue.*

Subject: *((smacks)) three bikinis ((swallows))*

She seems to be less influenced by the weight limit, at least concerning her speech alone.

Both dialogue snippets are examples of the *trouble* class, since both parts happen during the challenge dialogue stage.

4 RECOGNITION EXPERIMENTS

4.1 Pre-processing the Data

The acoustic feature set consists of the Emobase feature set, fully described in [EWS10], which is widely used for emotion recognition. The features are extracted using openSMILE [EWGS13]. The feature set includes 988 acoustic features extracted on utterance-level, such as intensity, loudness, 12 MFCCs, F_0, voicing probability F_0 envelope, 8 line spectral frequencies, zero-crossing rate, and their functionals. Other feature sets widely employed for emotion recognition, such as those described in [SSB09] [SSB+11] were tested in a preliminary investigation, but were rejected since they lead to poor results.

The physiological features are extracted from the biosignal data on dialogue stage level (including the speaking time of the Wizard), using the Augsburg Biosignal Toolbox[2]. The 3 original biosignals (EMG, RSP, SC) are preprocessed by applying a lowpass filter and normalisation, then a total number of 104 features, including first and second order derivatives and statistical features (mean, median, standard deviation, etc.) are calculated at a sampling rate of 32 Hz. The full description of the feature set can be found in [Wag09].

[2] http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/projects/tools/aubt/

4.2 Classification

We chose random forest as a classifier for the classification of both, acoustic and biosignal data. This classification method was chosen because of its higher training speed and its good performance compared to support vector machines [LW02], the standard classification method widely used for emotion recognition from speech. We employ the Weka implementation of random forest, which is based upon the classic algorithm by Breiman [Bre01]. One advantage of this implementation is that there are only two parameters to be tuned: the number of features used in each node and the number of trees. The hyperparameter optimisation takes place using grid search. For both types of data, between 1 and 50 features and between 10 and 100 trees are evaluated using the development subsets. For the acoustic data, the best parameters are found to be 6 features and 30 trees. For the biosignal data, the best parameters are found to be 3 features and 10 trees.

5 RESULTS AND DISCUSSION

The recall, precision and f-measure of the classification for the two classes of *trouble* and *baseline* are shown in Table 3 and Table 4 for the acoustic and biosignal data, respectively. A comparison of the UAR values for both types of data can be seen in Fig. 1.

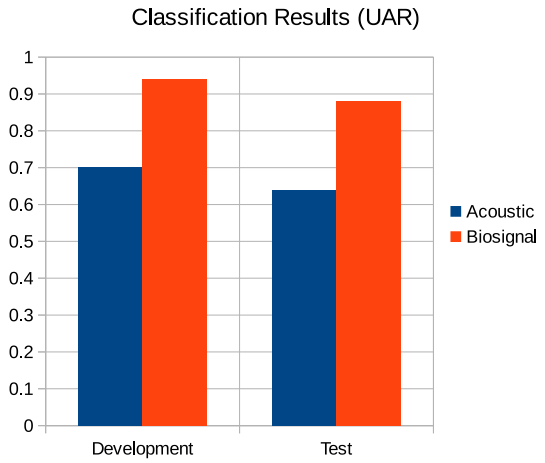


Figure 1: Comparison of classification results on physiological and acoustic data, UAR

On acoustic data, the UAR lies at 0.70 for the development set and 0.64 for the test set, with higher recall values for the *trouble* class and lower values for the *baseline* class. On biosignal data, the results are better by roughly 25 percentage points: the UAR lies at 0.94 for the development set and 0.88 for the test set, here the *trouble* class is recognised with a higher recall compared to the *baseline* class on the development set, but with a lower recall on the test set. But overall we can

see that the results on the test set are similar to the results on the development set, indicating that the model is able to appropriately generalise.

Regarding the precision of the detection we can see a comparable trend as for UAR. For the acoustic data, the unweighted average precision lies at 0.68 and 0.64 for the development and the test sets, respectively. For the biosignal data, the values of unweighted average precision are 22 percentage points higher for the development set and even 30 percentage points higher for the test set, resulting in 0.90 for the development set and an even higher value of 0.94 for the test set.

	Recall	Precision	F-Measure
Development Set			
Trouble	0.77	0.51	0.61
Baseline	0.63	0.84	0.72
Unweighted av.	0.70	0.68	0.67
Test Set			
Trouble	0.66	0.58	0.62
Baseline	0.62	0.70	0.66
Unweighted av.	0.64	0.64	0.64

Table 3: Classification results on acoustic data.

	Recall	Precision	F-Measure
Development Set			
Trouble	1.00	0.80	0.89
Baseline	0.88	1.00	0.93
Unweighted av.	0.94	0.90	0.91
Test Set			
Trouble	0.75	1.00	0.86
Baseline	1.00	0.89	0.94
Unweighted av.	0.88	0.94	0.90

Table 4: Classification results on physiological data.

Overall we can say that *trouble* can be recognised in both, the biosignal and the acoustic data, but the classification on the biosignal data clearly outperforms the classification on the acoustic data. This can be explained by the fact that, as already mentioned, the emotions contained in voice are easy to conceal, in contrast to the physiological reactions, which cannot be controlled deliberately.

Comparing our results to those found in the literature, we can say that the results on acoustic data are not as good as presented in [BFH+03], where an average recall of over 73% for a WOZ scenario and a two class problem (prosodic peculiarities vs. no prosodic peculiarities during a challenging dialogue) was reached. But as already mentioned, the data used there had a more detailed annotation, including POS tagging and, more importantly, annotations of prosodic peculiarities detected by the annotators, and not only annotations of dialogue stages supposed to lead to trouble, as in our case. Concerning biosignals, we achieve better results than the results described in [Kim07]. In a comparable WOZ setting including four levels on the valence-arousal scale, only 55% accuracy in subject-independent evaluation combining acoustic and biosignal data can be achieved there.

In general, we can say that recognising challenging stages of dialogues using biosignal data is reliable: even in this subject-independent evaluation we can recognise the *trouble* class with a very high certainty - we found 75% of all instances of the *trouble* class, with 0% false alarm rate. Unfortunately, the same cannot be said for using the acoustic data. For this case, we found only 58% of the instances, and only 70% of the found instances were indeed instances of the *trouble* class.

Although the results for the biosignal data are very promising, we have to consider that this data, in contrast to acoustic data, is not easily obtainable, especially EMG and RSP. We can assume that the compliance of human-computer interaction systems might suffer from intrusiveness of physiological sensors (here intrusiveness means constraints to the observed human). On the other hand, there are also easily obtainable types of physiological data, such as pulse and skin temperature, which can be collected from interaction devices like smartwatches etc. For further investigations, it would be interesting to focus on these easily obtainable types of physiological data.

One probable explanation for the different results on acoustic and biosignal data is that, as already mentioned, acoustic data can be easily manipulated by the subject. It is imaginable that some of the subjects forced themselves to speak calmly, since they were speaking to a computer. But in contrast to voice, the physiological reactions cannot be deliberately manipulated, and therefore more differences between the dialogue stages can be found and thus automatically detected. This also means, that the biosignal data can be used as "ground truth" to detect changes in human emotional state that cannot be detected from speech, and also to annotate them. But, on the other hand, trouble recognition using only acoustics also should not be ignored: for tasks where no data other than acoustic data is available we can still detect over 60% of challenging dialogue stages using our approach. One of such tasks could be call center applications, where a trouble detection system could support human call center agents [SO15].

Another problem concerning emotion recognition from speech is that there is still no consensus in the litera-

ture regarding which features are best suited for this difficult task – it might be that the usually employed features do not represent the differences between various emotions. Additionally, many feature extraction routines base on human perception models – but, as already mentioned before, emotion recognition cannot be done with a 100% accuracy by human annotators. This also opens the question of gathering the "golden standard" – to build the right model for emotion recognition, we need to ensure that the emotions are labelled correctly in the data. A widely employed but costly solution for this problem is to obtain annotations from multiple raters and to use only data with a high inter-rater agreement, which, however, is also difficult to achieve [SBW14]. Our results encourage to rely on biosignal data as ground truth instead, therefore saving the effort of multiple annotation procedures.

6 CONCLUSION

In this paper, we investigated how challenging dialogue stages in naturalistic human-computer interaction can be automatically recognised. For this task, we used the recordings of the LAST MINUTE Corpus. The recordings include non-challenging and challenging parts of WOZ human-computer interaction, which were consolidated into two classes: the *baseline* class and the *trouble* class. Instead of widely employed support vector machines we used random forest for this classification task. We achieved an UAR of 64% on acoustic data and an UAR of 88% on biosignal data, showing that it is possible to detect challenging parts of an interaction using acoustic data as well as physiological data. However, we did not perform human perception tests regarding the levels of trouble audible in the acoustic data and used only simple annotations.

There are two main directions for future research. First, it should be investigated, whether the recognition rate can be improved by more complex annotations of different levels of trouble. As mentioned before, different subjects may experience and express different levels of trouble. An important question is whether age, sex or other factors influence the experienced and expressed level of trouble during a challenging human-computer interaction. If this is the case, using different models for different user groups should improve the results.

Another direction for future work is to exploit the multimodality of the data, using both acoustic and biosignal data simultaneously, since it was already proven in the literature that multimodal approaches can improve the detection results [SSM15]. Especially combinations of acoustics and easily obtainable biosignals like pulse could be interesting for this task. Unfortunately, a multimodal investigation was not possible in this setting because of missing synchronisation of the used data sets. We will approach this problem in future research.

7 ACKNOWLEDGEMENT

The work presented in this paper was done within the Transregional Collaborative Research Centre SFB/TRR 62 'Companion-Technology for Cognitive Technical Systems' (`www.sfb-trr-62.de`) funded by the German Research Foundation (DFG). The first author was additionally funded by the consortium 3Dsensation (`www.3d-sensation.de/`), a part of the Zwanzig20 German government funding program.

8 REFERENCES

[BBJ14] Böck, R., Bergmann, K., and Jaecks, P. Disposition recognition from spontaneous speech towards a combination with co-speech gestures. In *Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction*. Springer, 2014, pp. 57–66.

[BDY+04] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th ICMI* (2004), ACM, pp. 205–211.

[BFH+03] Batliner, A., Fischer, K., Huber, R., Spilker, J., and Nöth, E. How to find trouble in communication. *Speech Communication 40* (2003), 117–143.

[BPR+05] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., and Weiss, B. A database of german emotional speech. In *Proceedings of the INTERSPEECH-2005* (Lisbon, Portugal, 2005), pp. 1517–1520.

[Bre01] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[CBL+00] Cacioppo, J. T., Berntson, G. G., Larsen, J. T., Poehlmann, K. M., and Ito, T. A. The psychophysiology of emotion. *Handbook of emotions 2* (2000), 173–191.

[CW05] Choi, A., and Woo, W. Physiological sensing and feature extraction for emotion recognition by exploiting acupuncture spots. In *Affective Computing and Intelligent Interaction*. Springer, 2005, pp. 590–597.

[EFE72] Ekman, P., Friesen, W. V., and Ellsworth, P. *Emotion in the human face: Guidelines for research and an integration of findings.* Pergamon Press, New York, 1972.

[ELF83] Ekman, P., Levenson, R. W., and Friesen, W. V. Autonomic nervous system activity distinguishes among emotions. *Science 221*, 4616 (1983), 1208–1210.

[EWGS13] Eyben, F., Weninger, F., Gross, F., and Schuller, B. Recent developments in opensmile,

the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM International Conference on Multimedia* (New York, NY, USA, 2013), MM '13, ACM, pp. 835–838.

[EWS10] Eyben, F., Wöllmer, M., and Schuller, B. openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *Proc. of the ACM MM-2010* (Firenze, Italy, 2010), pp. 1459–1462.

[FMR+12] Frommer, J., Michaelis, B., Rösner, D., Wendemuth, A., Friesen, R., Haase, M., Kunze, M., Andrich, R., Lange, J., Panning, A., and Siegert, I. Towards emotion and affect detection in the multimodal last minute corpus. In *Proceedings of the 8th LREC* (Istanbul, Turkey, 2012), pp. 3064–3069.

[HGSW04] Haag, A., Goronzy, S., Schaich, P., and Williams, J. Emotion recognition using bio-sensors: First steps towards an automatic system. In *Affective Dialogue Systems* (2004), Springer, pp. 36–48.

[KBK04] Kim, K. H., Bang, S., and Kim, S. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing 42*, 3 (2004), 419–427.

[Kim07] Kim, J. Bimodal emotion recognition using speech and physiological changes. In *Robust Speech Recognition and Understanding*, M. Grimm and K. Kroschel, Eds. I-Tech Education and Publishing, Vienna, Austria, 2007, pp. 265–280.

[LEF90] Levenson, R. W., Ekman, P., and Friesen, W. V. Voluntary facial action generates emotion-specific autonomic nervous system activity. *Psychophysiology 27*, 4 (1990), 363–384.

[LW02] Liaw, A., and Wiener, M. Classification and regression by randomforest. *R news 2*, 3 (2002), 18–22.

[Meh96] Mehrabian, A. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology 14* (1996), 261–292.

[PRS+14] Prylipko, D., Rösner, D., Siegert, I., Günther, S., Friesen, R., Haase, M., Vlasenko, B., and Wendemuth, A. Analysis of significant dialog events in realistic human-computer interaction. *Journal on Multimodal User Interfaces 8* (2014), 75–86.

[RAE+14] Ringeval, F., Amiriparian, S., Eyben, F., Scherer, K., and Schuller, B. Emotion recognition in the wild: Incorporating voice and lip activity in multimodal decision-level fusion. In *Proceedings of the 16th ICMI* (2014), ACM, pp. 473–480.

[RBND06] Rainville, P., Bechara, A., Naqvi, N., and Damasio, A. R. Basic emotions are associated with distinct patterns of cardiorespiratory activity. *International journal of psychophysiology 61*, 1 (2006), 5–18.

[RFF+12] Rösner, D., Frommer, J., Friesen, R., Haase, M., Lange, J., and Otto, M. LAST MINUTE: a Multimodal Corpus of Speech-based User-Companion Interactions. In *Proceedings of the 8th LREC* (Istanbul, Turkey, 2012), pp. 96–103.

[SBW14] Siegert, I., Böck, R., and Wendemuth, A. Inter-rater reliability for emotion annotation in human-computer interaction: comparison and methodological improvements. *Journal on Multimodal User Interfaces 8*, 1 (2014), 17–28.

[Sch81] Scherer, K. R. Speech and emotional states. *Speech evaluation in psychiatry* (1981), 189–220.

[SDS+13] Sikka, K., Dykstra, K., Sathyanarayana, S., Littlewort, G., and Bartlett, M. Multiple kernel learning for emotion recognition in the wild. In *Proceedings of the 15th ICMI* (2013), ACM, pp. 517–524.

[SO15] Siegert, I., and Ohnemus, K. A new dataset of telephone-based human-human call-center interaction with emotional evaluation. In *Proceedings of the 1st International Symposium on Companion Technology* (Ulm, Germany, September 2015), pp. 143–148.

[SSB09] Schuller, B., Steidl, S., and Batliner, A. The INTERSPEECH 2009 Emotion Challenge. In *Proceedings of the INTERSPEECH-2009* (Brighton, UK, 2009), pp. 312–315.

[SSB+11] Schuller, B., Steidl, S., Batliner, A., Schiel, F., and Krajewski, J. The INTERSPEECH 2011 Speaker State Challenge. In *Proceedings of the INTERSPEECH-2011* (Florence, Italy, 2011), pp. 3201–3204.

[SSM15] Schwenker, F., Scherer, S., and Morency, L. Multimodal pattern recognition of social signals in human-computer-interaction. *Lecture Notes in Computer Science 8869* (2015).

[SVE+09] Schuller, B., Vlasenko, B., Eyben, F., Rigoll, G., and Wendemuth, A. Acoustic emotion recognition: A benchmark comparison of performances. In *Proceedings of the IEEE ASRU-2009* (Merano, Italy, 2009), pp. 552–557.

[Wag09] Wagner, J. *The Augsburg biosignal toolbox*. University of Augsburg, 2009.

[YH08] Yannakakis, G. N., and Hallam, J. Entertainment modeling through physiology in physical play. *International Journal of Human-Computer Studies 66*, 10 (2008), 741–755.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Infrared-based Object Classification for the Surveillance of Valuable Infrastructure

Christos Palaskas[1] Savvas Rogotis[1] Dimos Ioannidis[1,2] Dimitrios Tzovaras[1] Spiros Likothanassis[2]

[1]Information Technologies Institute – Centre for Research and Technology Hellas
57001,Thermi,Thessaloniki, Greece
{chris.palaskas,srogotis,djoannid, dimitrios.tzovaras}@iti.gr

[2]Pattern Recognition Laboratory – Computer Engineering and Informatics – University of Patras
26500,Rio, Patras, Greece
{djoannid,likothan}@ceid.upatras.gr

ABSTRACT

The surveillance of valuable infrastructure, such as photovoltaic parks, is considered of fundamental importance for their proper function and maintenance as well as the avoidance of criminal damage incidents. At the same time, the privacy of employees working in the same area should not be jeopardized and their personal data should always be protected. The use of thermal cameras presents a solution to both of the above issues by offering an unobtrusive surveillance approach with the ability to supervise industrial premises under a wide range of environmental and situational conditions. The current paper proposes an algorithm for the classification of moving objects that aims to increase the efficiency of surveillance methodologies by shifting the focus on high-risk classes, such as humans instead of animals. The proposed methodology utilizes an automated decision framework that determines when textural features are fit to be used, based on the discriminative power of the texture of the object. Many texture descriptors were tested, including Local Phase Quantisation and Histograms of Oriented Gradients, resulting in the use of a lately proposed combination of these descriptors. This new multi-class object classification approach introduces the use of confidence values and a voting system to achieve a more accurate selection of the appropriate class. The velocity was also used as a discriminative feature, especially to help distinguish between humans and motorcycles. Several algorithms have been used to validate the results of the experimental studies with special focus on the classification accuracy. The experimental results were obtained from a series of scenarios demonstrated in four different condition sets (different temperature-humidity-illumination), that exposes the advantages and disadvantages of the proposed unimodal classification method in infrared imagery. The dataset is also benchmarked against another state-of-the-art approach.

Keywords

Thermal Imaging; Multi-Class Classification; Shape Descriptor; Texture Descriptor; Local Phase Quantization; Histogram of Oriented Gradients; Contour Point Distribution; Surveillance

1. INTRODUCTION

Monitoring and surveilling facilities and estates has been a primary need of human society since the dawn of time. There is a number of ways to obtain information to support the protection of a facility, such as color cameras, depth cameras, thermal cameras, noise sensors, CO_2 emission sensors and so on. Over the last few years, an explosion of camera invasion in everyday life was witnessed: streets, airports, hospitals, shopping malls, office buildings are loaded with cameras, surveilling and tracking anything that moves within their field of view. This camera-ubiquity amplified the necessity for more privacy preserving techniques, but without compromising the level of security. Such a technology is the monitoring of facilities using infrared cameras, since no visual features of the human face are obtained. Although there are some disadvantages using such a camera instead of a visual one, such as the low resolution of IR images and the high market price, there are also advantages, such as the ability to operate even at challenging conditions (e.g. nighttime, through fog, smoke). The use of such technology has released lots of human resources in the field of security and surveillance,

replacing guards with intelligent surveillance systems.

The main input instrument in the process of surveilling, the camera, can function in various wave ranges: optical, long wave infrared, short wave infrared, medium wave infrared etc. Many works have been published dealing with the fusion of more than one modes of input. In general, the tradeoff between optical imagery and thermal – IR imagery is that the first gives more information in specific environmental sets but almost no information in other sets, i.e. optical imagery works only with optical light but yields no information in darkness, whereas IR imagery, which depends on the emission of IR light due to the temperature of the bodies, is not affected by light, making it a more efficient method to monitor during nighttime.

This paper presents an improved method for classifying moving objects from infrared images into 4 classes, namely human, car, motorcycle and animal, using spatial features (shape, texture) and temporal (velocity) and feeding them to a voting system. The textural features are used only when the image of the object has enough colour discrimination, otherwise the classification is based only on the shape and velocity. The algorithm has been tested in multiple and demanding weather sets and illumination conditions. The features (shape and texture) have been selected based on performance, after testing and discarding many other features. Also the proposed classifier has been tested using the same features on another classifier. Finally the whole approach was benchmarked against another state-of-the-art algorithm, using the same challenging dataset with encouraging results.

The remainder of the paper is organized as follows: Section 2 reports some of the related work in the field. In Section 3 the implemented method is described, namely the feature extraction of the object, the training of the model and the final classification process. Experiments and discussions are provided in Section 4, while Section 5 concludes the paper.

2. RELATED WORKS

Effective monitoring and surveillance is prerequisite for adequate object detection [Jo13a]. Other ways are background subtraction [Bar11a] [Van12a], or segmentation [Arb11a] [Alp12a] followed by tracking [Bab11a] within the camera's field. As the object has been discriminated from the background it is feasible to extract its features and move on to its classification.

Many works dealing with classification methods have been published, but most of them concern the optical input mode (RGB and grayscale). In one of these works [Lia15a] a two-level Haar wavelet transform is applied to the bounding window of the object in an RGB colored image, and from these two level bands the local shape features and the Histogram of Oriented Gradients (HOG), are extracted. These are fed to a Support Vector Machine (SVM) which has been trained from a data set, so as to classify the object into one of four classes (human, bicycle, motorcycle, car).

There have also been some approaches [Ku10a] [Shu11a] where only the shape was used to classify objects in infrared imagery. On the other hand, many works use only textural descriptors, as will be presented below, but nothing significant was found using both and also determining when it is profitable to do so, as is the focus of this work.

In a recent work which aimed at detecting pedestrians at night [Joh15a] an adaptive fuzzy C-means clustering was adopted to segment the IR images and retrieve the candidate pedestrians. A convolutional neural network was then used to simultaneously learn relevant features and perform the binary classification.

In another recent work [Wan15a] a spatiotemporal saliency model based on three-dimensional Difference-of-Gaussians filters was proposed for small moving object detection in infrared videos. First, instead of utilizing the spatial Difference-of-Gaussians (DoG) filter which has been used to build saliency models for static images, they proposed the extension of the spatial DoG filter to construct three-dimensional (3D) Difference-of-Gaussians filters for measuring the center-surround difference in the spatiotemporal receptive field. After that, an effective spatiotemporal saliency model was generated based on those filters.

Attempting to detect humans in infrared imagery using a gradient-based technique, the authors in [Olm12a] introduced the exploitation of local information histogram of orientations of phase coherence. Thus, they obtained a scale, brightness and contrast invariant descriptor that can detect pedestrians in IR images. In another work [Che12a] the geometrical-based features and the texture-based features are extracted and then are fed to two trained classifiers: the kNN and the SVM. The geometrical features used are: aspect ratio, compactness, fill ratio, and Hu's invariant moments [Hua10a]. The textural features are: smoothness, uniformity and skewness (a metric that indicates where the bulk of the distribution histogram lies, i.e. to the left or to the right and to what extent. The features are range-normalized and then a Principal Component Analysis (PCA) is performed, to maintain the most significant features of the classification.

A shape-based fuzzy network [Jua08a] has also been used for moving object classification. The distance of the center of the object to every point of the contour is calculated and smoothed, and the coefficients obtained from their discrete Fourier transform are used for the feature vector, and so is the aspect-ratio. The feature vector in its turn is used to construct a Self-cOnstructing Neural Fuzzy Inference Network (SONFIN) used for object recognition.

In another approach [Asp14a] the object was divided into ringlets, each one contributing to the creation of a histogram. The value of each pixel was weighted according to a Gaussian distribution of the distance from the ringlet. This feature turned out to be rotation invariant and centered weighted, since the significance of the ringlets closer to the center are more important than the outer ones in some cases.

All said, there seems to be a difficulty in the discrimination between humans and motorcycles / bicycles using shape descriptors, because the top half of both classes is identical, and many times in infrared imagery there is a partial detection. There also seems to be an inadequacy in the decision of the use of shape and/or texture descriptors robustly. This would provide better classification results, aiding the surveilling purpose, by allotting more resources to the surveillance of particular classes, which are considered a higher threat. For example, a human poses more of a threat than a dog in an outdoor surveillance system.

3. METHODOLOGY

In an effort to better distinguish humans from motorcycles, a new algorithm for the classification of moving objects in infrared imagery into multiple classes is proposed, using both textural and shape descriptors, along with the velocity of the object. The result of the algorithm from every frame of the sequence must be used with a tracker in a voting system, allowing for greater accuracy. An object detection algorithm based on ViBE [Bar11a] [Van12a] and a moving object tracker based on the work proposed on [Tor12a] have been implemented for the first stage of surveillance, prior to classification, but their results are not being discussed, as their interest lies out of the purpose of this work. Every ROI from the segmentation is tracked even when it stops for a while. When two or more objects enter the scene they are tracked and classified separately, except the case when they come extremely close. In this case classification stops, and continues only if the objects separate again. When this happens, another texture based classification takes place, in order to give them the right IDs, the ones that were provided before the merge. After this, object classification continues normally.

3.1 Feature Extraction

The features selected for the classification process regard the following categories: Shape descriptors, Texture descriptors and Velocity. The features were selected according to their ability to describe objects of the same class (small intraclass variation) and at the same time differentiate objects from different classes (large interclass variation). A 3-D representation of the dispersion of the training set after multidimensional scaling (MDS) can be seen in Figure 1 and Figure 2. These features were selected after testing the discriminative power of many other features, and particularly horizontal and vertical projections [Gur11a], Gaussian Ringlet Intensity Distribution features [Asp14a], aspect ratio, fill ratio, uniformity, skewness, smoothness, compactness, Hu's moments [Hu62a], Local Binary Patterns (LBP) [Oja02a], Histogram of Oriented Gradients (HOG) [Tsa10a] and Local Phase Quantization (LPQ) [Jia14a].

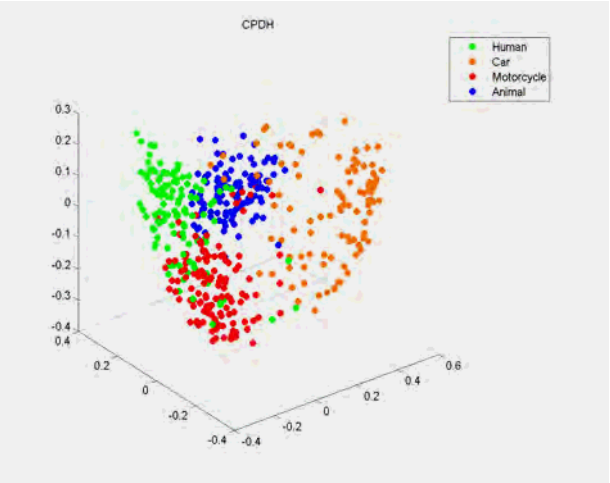


Figure 1. Dispersion of the four classes after MDS using the CPDH shape descriptor

3.1.1 Shape Descriptors

The description of the shape of the object is performed using the Contour Points Distribution Histogram (CPDH) algorithm [Gur11a], which is implemented by measuring the distribution of the contour points in a predefined topography.

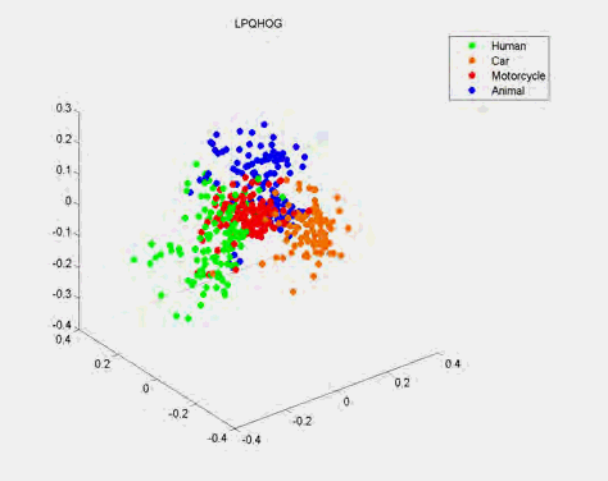


Figure 2. Dispersion of the four classes after MDS using the LPQHOG texture descriptor

The object is divided into 36 segments – 3 zones around the object by twelve 30 degree sectors. By counting the points of the contour that belong to each segment, the histogram of the distribution is produced (Figure 3). This histogram is used as a shape descriptor feature. Its discriminative ability is shown in the distance matrix of the training set in Figure 5, where the 4 classes of 130 objects each, are distinguished adequately. Blue represents zero (0) distance that goes towards red which represents one The main diagonal is of course blue, which means that every histogram has zero distance from itself. The distance between objects of the same class is small (blue), while it is large (red) between objects of different classes. The similarity between the first and third class (human / motorcycle) can be distinguished using the velocity feature. (1).

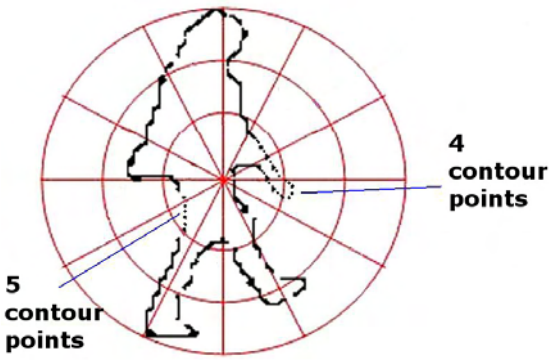


Figure 4. Contour Points Distribution into geometric segments.

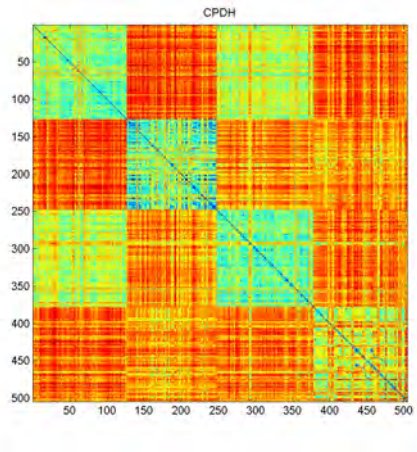


Figure 5. Distance matrix of training data with CPDH histograms

3.1.2 Texture Descriptors

The Local Phase Quantization histogram concatenated with the Histogram of Oriented Gradients first introduced in [Rog15a] was used as a texture descriptor. The gradient based properties of the HOG descriptor are enriched with information derived from the frequency domain of the LPQ descriptor with better results than each descriptor on its own.

The LPQ descriptor is performed in local areas where quantization of the Fourier transform phase takes place. Phase information is extracted using a 2-D Discrete Fourier Transform extracted from a rectangular N-by-N neighborhood N_x on every pixel x of the image $f(x)$ defined by

$$F(\mathbf{u},\mathbf{x}) = \sum_{\mathbf{y}\in N_x} f(\mathbf{x}-\mathbf{y}) e^{-j2\pi \mathbf{u}^{\mathrm{T}}\mathbf{y}} = \mathbf{w}_{\mathbf{u}}^{\mathrm{T}}\mathbf{f}_{\mathbf{x}} \qquad (1)$$

where $\mathbf{w_u}$ is the basis vector of the 2-D DFT at frequency $\mathbf{u}$, $\mathbf{f_x}$ is the vector containing all N^2 samples from N_x, and j the imaginary unit (j^2=-1). The local Fourier coefficients are computed at four frequency points: $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u^3 = [a, a]^T$, *and* $u_4 = [a, -a]^T$, where a is a sufficiently small scalar (a = 1/7 in our implementation), and T the transpose of a vector, which is denoted by a bold symbol.

The histogram of oriented gradients is a powerful texture descriptor created by calculating the gradient values of the pixel's intensity using a discrete derivative mask in horizontal and vertical direction, or even the sum of the vectors in all directions (Figure 5). The vector is the difference in intensity and direction between the central pixel and its neighbor. As such, the feature captures the

distribution of intensity gradients within the object of interest and expresses it by a single histogram.

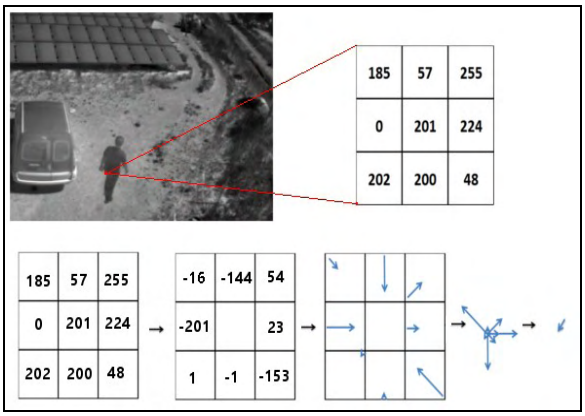


Figure 5. Gradient extraction of a pixel

The discriminative power of the texture descriptor is shown in Figure 6. In general, texture variation is limited in infrared imagery in comparison to RGB images, which explains why the shape descriptor yields better results. But it will be shown that the use of both texture and shape descriptors yields better results.

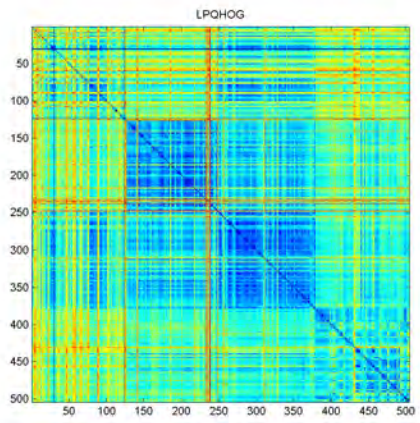


Figure 6. Distance matrix of training data with LPQHOG histograms

3.1.3 Velocity

In order to distinguish between humans and motorcycles and between cars and animals the velocity of the object can be used as a feature. For a mounted camera on a pole the objects appear larger and moving faster as they approach, while they appear smaller and moving slower when they are far from the camera. It was found that the ratio of the distance in pixels per frame over the height of the object is proportional to the actual speed of the object, so it was used as a feature. The velocity is calculated by dividing the Euclidean distance of the center of mass of the object between n frames by the height of the object:

$$u_i = \frac{\|C_i - C_{i-n}\|}{h_i} \quad (2)$$

u_i is the velocity of the object at frame i, C_i is the center of mass of the moving object at frame i, and h_i is the height of the object at frame i.

Figure 7. Examples of the templates of the training database

3.2 Training – Database description

The training dataset (Figure 7) was collected during a span of more than half a year so that images from very different environmental conditions would be stored, using a stationary long-wave infrared camera providing 320X256 pixel images. During the data collection, about 31,600 images of humans were obtained, 15,400 car images, 6,400 motorcycle images and 4,400 images of dogs. The images were obtained during day and night, during sunny, cloudy and rainy days, in hot and cold weather. Different types of cars and motorcycles were used during the course of collecting the dataset. Images of humans were captured in various poses, and wearing a variety of clothes. The predescribed features were extracted from all the obtained images and two SVM classifiers were created. Finally, the training set was refined, selecting one hundred and thirty images from each class as training data, based on the distance of their features in the SVM space, both textural and shape based.

Another database was recorded for the experimental evaluation of the proposed algorithm, called testing dataset. The following scenarios were repeated in four different weather conditions, i.e. early in the morning, on a sunny afternoon, at night and on a rainy morning.

1. Human walks and runs
2. Car drives
3. Motorcycle drives
4. Animal wanders

About 15,000 images were recorded under all weather conditions, with a total of 62,784 images.

3.3 Classification

From every object that is recognized as foreground in a frame, the shape descriptor and the velocity described above are extracted and projected on the feature space of the SVM. The velocity is the last dimension of the SVM hyperspace. The distance to the nearest support vector of the shape descriptor classifier is calculated for every class. After that, the image's median is calculated to decide whether its textural data will help discriminate its class. The optimal value (med_o) was determined by testing the accuracy of the algorithm on a wide range of values, using only images from the testing dataset that were pertinent to the experiment, i.e. that had small textural discrimination (Figure 9). For images with enough textural data the texture descriptor described above is extracted and projected on the feature space of the SVM. (Figure 8)

Therefore, four or eight distances are calculated for every object: four are the distances of the object to each class' boundary in the SVM hyperspace of the shape descriptors and the velocity and four are the corresponding distances of the texture descriptor. All distances are normalized with the minimum and maximum value of the corresponding training set, i.e. the distance of an object to the human shape descriptor SVM is normalized using the distances of all training data to the same SVM. These distances are considered as confidence values of the classifier: the greater the distance from the border of the two classes, the greater the confidence. The use of these confidence values was validated by running the entire testing dataset over a wide range of confidence level values (i.e. from zero to one, using a 0.05 step), and discarding the votes that had lower confidence than the value (Figure 10). The diagram is indicative of the intrinsic capability of the proposed algorithm to cope with results of low confidence, maintaining in this manner the overall accuracy of the algorithm at 0.83 by avoiding false negatives. Furthermore, a decrease in accuracy is only visible after the exclusion of solutions with confidence above 0.4 which affects the number of the false positives. So if an object that is very different from the four classes enters the scene, it will remain unclassified, as it's confidence value will be low.

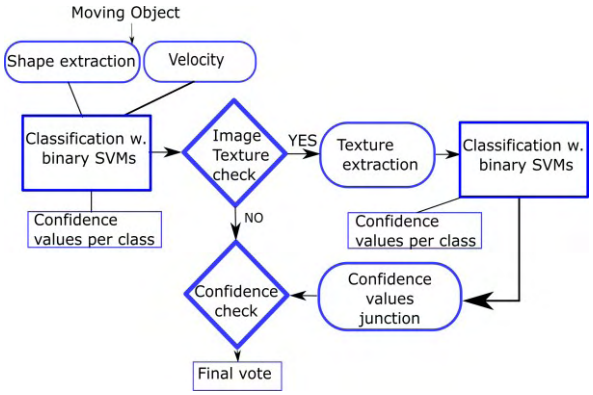


Figure 8. Classification procedure-block diagram

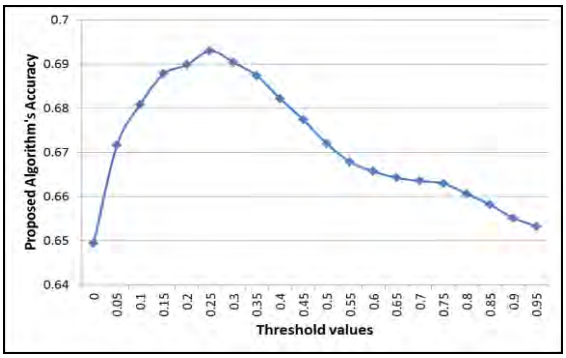


Figure 9. The accuracy of the algorithm over a range of texture discrimination values applied on a portion of the dataset with low texture images

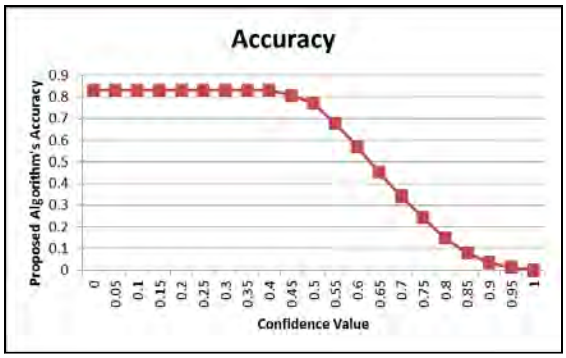


Figure 10. The accuracy of the algorithm over a range of confidence values applied on the whole dataset

In case the object's texture does not have enough discriminative power, which usually happens in infrared imagery when the object emits significantly more radiation than its background, only the shape descriptors and the velocity are used to classify the object. In all other cases, both shape/velocity and texture are used for the classification by fusing the results of the classifiers, by adding the normalized confidence value of each decision. The confidence

that the object belongs to the returned class is given by the formula:

$$C_{class} = \begin{cases} \max\begin{cases} C_{SH} \\ C_{SC} \\ C_{SM} \\ C_{SA} \end{cases}, & if\,(med > med_o) \\ \max\begin{cases} \frac{C_{SH}+C_{TH}}{2} \\ \frac{C_{SC}+C_{TC}}{2} \\ \frac{C_{SM}+C_{TM}}{2} \\ \frac{C_{SA}+C_{TA}}{2} \end{cases}, & else \end{cases} \quad (3)$$

where C_{XY} is the normalized distance of the X descriptor (Shape/Velocity or Texture) to the Y SVM (Human, Car, Motorcycle, Animal), *med* the median of the object and med_o the optimum value of the median, that was defined experimentally, as mentioned earlier.

The object is tracked in time and a vote is added to the class it is classified at every frame, if the confidence is above the confidence value. The object is finally classified into the class that has the most votes. For the purposes of this paper, and in order to address the possibility of inputs outside the spectrum of the datasets that were used for training and testing, a threshold of 0.25 was selected.

Figure 11. A human that can be discriminated through the use of texture descriptors

Figure 12. A dog in challenging condition where it fails to be distinguished from its surrounding background street

Figure 13. A human emitting significantly more radiation than the background

4. EVALUATION RESULTS - DISCUSSION

As was expected, using only shape descriptors did not yield encouraging results, especially in classifying humans from motorcycles, though it recognized cars and motorcycles adequately. (Table 1) This was a driving factor to the proposed work, which included textural descriptors, temporal features (velocity) and a decision on the use of textural features that improved significantly the classification results.

The proposed algorithm works well when the object's shape is well defined and there is textural discrimination, i.e. when the background emits more or less radiation than the object, but not excessively (Figure 11). When they emit the same IR radiation it is very difficult to differentiate between background and foreground (Figure 12). In case the object emits much more radiation than the background, only shape descriptors are used, lowering the accuracy of the decision (Figure 13).

The algorithm's accuracy has been estimated over the demanding testing dataset, and has an overall

accuracy of 83%. The precision and recall of the classifier on different weather sets are shown in Figure 14 and Figure 15. Unfortunately there was no data with animals during the rainy morning session. The precision for the Motorcycle class is low because in some cases it was mistakenly classified as human. This happened especially when the road was as hot as the motor, so the only part left distinguishable was the rider.

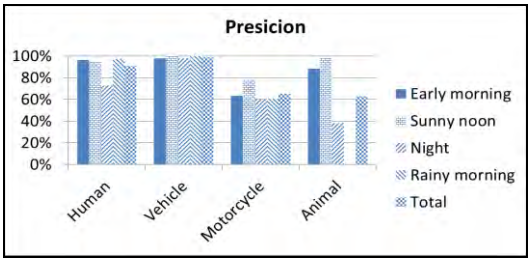


Figure 14. Precision per class

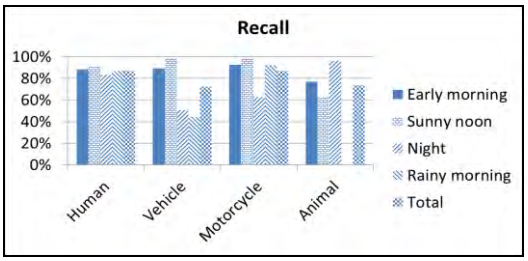


Figure 15. Recall per class

The descriptors were tested on the same data base (all weather conditions) using the Random Forest Classifier [Liv05a] (Table 2) instead of the proposed classifier (Table 3), but the results imply that the proposed classifier works better. The Random Forest Classifier scored 78.2% on accuracy. It was utilized in both feature sets, i.e. texture and shape providing two votes, although in some occasions, when the textural discrimination was low, it returned only one vote, based solely on the shape. The proposed algorithm does not return any votes if the confidence of the two binary classifiers is low.

The implemented method for benchmarking was found in the bibliography and was used to validate the proposed method using the same data set. In the work of [Wan10a] a Shape Context Descriptor is proposed where the log-polar histogram of the shape is exported in 5 bins for *log r* (radius) and 12 bins for θ (angle). Then a Shape context based Adaboost cascade classifier is used to classify the shape. The average time for a frame to be processed completely (segmentation, classification, tracking) using the proposed method is 46.9msec, while the benchmarking method needs 33.4msec. The times were computed on a Intel Core i7-4790K processor at 4.00GHz with 8GB RAM. The reason for this difference is that the proposed method performs almost the same tasks as the benchmark method, plus the texture feature extraction.

Wang's Algorithm		Actual classes: Human	Car	Moto	Animal
Predicted classes (votes)	Human	**299**	9	117	57
	Car	108	**1417**	44	146
	Moto	2486	30	**943**	16
	Animal	208	141	44	**166**
				Overall Accuracy	**45.3%**

Table 1. Benchmarking algorithm's accuracy

Proposed Algorithm		Actual classes: Human	Car	Moto	Animal
Predicted classes (votes)	Human	**2633**	140	132	10
	Car	0	**811**	11	0
	Moto	389	36	**954**	85
	Animal	19	138	0	**266**
				Overall Accuracy	**83%**

Table 2. Proposed algorithm's accuracy

Random Forest Classifier		Actual classes: Human	Car	Moto	Animal
Predicted classes (votes)	Human	**2047**	4	126	7
	Car	22	**1071**	5	5
	Moto	934	31	**964**	33
	Animal	38	19	2	**316**
				Overall Accuracy	**78.2%**

Table 3. Accuracy of proposed method using random forest classifiers instead of SVMs

5. CONCLUSIONS

This paper introduced a new approach in multi-class object classification, using confidence values in each decision, and deciding whether to use textural

features along with a shape descriptor and velocity on each frame. In the highly demanding field of infrared thermography, there must be enough flexibility to choose the best features to use for classification, either texture and shape, or only shape. The algorithm reached an overall accuracy of 83%, but can climb up to 91% under certain climate conditions (e.g. a sunny afternoon). The structure of the approach allows for many improvements in future works: from choosing different descriptors to replacing the binary classifiers with other methods. The field is open for more research especially regarding the extraction of features from objects that are partially detected due to heavy occlusion. The size of the testing and training dataset may also grow in future work to allow for greater validity of the results and also for more discussion.

6. ACKNOWLEDGEMENTS

This work was partially supported by the EU funded PREACT Capability Project (CP) (FP7-607881).

7. REFERENCES

[Alp12a] Alpert, Sharon, et al. "Image segmentation by probabilistic bottom-up aggregation and cue integration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 34.2, p.315-327, (2012).

[Arb11a] Arbelaez, Pablo, et al. "Contour detection and hierarchical image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 33.5, p.898-916, (2011).

[Asp14a] Aspiras, Theus H., Vijayan K. Asari, and Juan Vasquez. "Gaussian ringlet intensity distribution (GRID) features for rotation-invariant object detection in wide area motion imagery", IEEE International Conference on Image Processing (ICIP), p.2309-2313, (2014).

[Bab11a] Babenko, B., Ming-Hsuan Y., and Belongie, S., "Robust object tracking with online multiple instance learning", IEEE Transactions on Pattern Analysis and Machine Intelligence 33.8, p.1619-1632, (2011).

[Bar11a] Barnich, Olivier, and Marc Van Droogenbroeck. "ViBe: A universal background subtraction algorithm for video sequences", IEEE Transactions on Image Processing, 20.6, p.1709-1724, (2011).

[Che12a] Chen, Eli, Oren Haik, and Yitzhak Yitzhaky. "Classification of moving objects in atmospherically degraded video", Proceedings of SPIE-The International Society for Optical Engineering, 51.10, p.1-14, (2012).

[Gur11a] Gurwicz, Yaniv, Raanan Yehezkel, and Boaz Lachover. "Multiclass object classification for real-time video surveillance systems", Elsevier, Pattern Recognition Letters 32.6, 805-815, (2011).

[Hu62a] Hu, Ming-Kuei. "Visual pattern recognition by moment invariants", IRE Transactions on Information Theory, 8.2, p179-187, (1962).

[Hua10a] Huang, Zhihu, and Jinsong Leng. "Analysis of Hu's moment invariants on image scaling and rotation", IEEE 2nd International Conference on Computer Engineering and Technology (ICCET), 7, p.476-480, (2010).

[Jia14a] Jiang, Bihan, et al. "A dynamic appearance descriptor approach to facial actions temporal modeling", IEEE Transactions on Cybernetics, 44.2, p.161-174, (2014).

[Jo13a] Jo, Ahra, et al. "Performance improvement of human detection using thermal imaging cameras based on mahalanobis distance and edge orientation histogram", Information Technology Convergence, Springer Netherlands, Lecture Notes in Electrical Engineering, 253, p.817-825, (2013).

[Joh15a] John, Vijay, et al. "Pedestrian detection in thermal images using adaptive fuzzy C-means clustering and convolutional neural networks", IEEE International Conference on Machine Vision Applications (MVA), 14th IAPR, p246-249, (2015).

[Jua08a] Juang, Chia-Feng, and Liang-Tso Chen. "Moving object recognition by a shape-based neural fuzzy network", International Conference on Artificial Neural Networks (ICANN 2006) / International Conference on Engineering of Intelligent Systems (ICEIS 2006), Neurocomputing 71.13, p2937-2949, (2008).

[Ku10a] Ku, Zhi Kai, Chee Fei Ng, and Siak Wang Khor. "Shape based recognition and classification for common objects-An application in video scene analysis", IEEE 2nd International Conference on Computer Engineering and Technology (ICCET), 3, p13-16, (2010).

[Lia15a] Liang, Chung-Wei, and Chia-Feng Juang. "Moving object classification using local shape and HOG features in wavelet-transformed space with hierarchical SVM classifiers", Applied Soft Computing 28, p483-497, (2015).

[Liv05a] Livingston, Frederick. "Implementation of Breiman's random forest machine learning algorithm", ECE591Q Machine Learning Journal Paper (2005).

[Oja02a] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", IEEE Transactions on Pattern

Analysis and Machine Intelligence, 24.7, p971-987, (2002).

[Olm12a] Olmeda, Daniel, Arturo de la Escalera, and Jose Maria Armingol. "Contrast invariant features for human detection in far infrared images", IEEE Intelligent Vehicles Symposium (IV), p117-122, (2012).

[Rog15a] Rogotis, Savvas, et al. "Recognizing suspicious activities in infrared imagery using appearance-based features and the theory of hidden conditional random fields for outdoor perimeter surveillance", Journal of Electronic Imaging 24.6, p.1-10, (2015).

[Shu11a] Shu, Xin, and Xiao-Jun Wu. "A novel contour descriptor for 2D shape matching and its application to image retrieval", Image and Vision Computing 29.4, 286-294, (2011).

[Tor12a] Torabi, Atousa, Guillaume Massé, and Guillaume-Alexandre Bilodeau. "An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications", Computer Vision and Image Understanding 116.2, 210-221, (2012).

[Tsa10a] Tsai, Grace. "Histogram of oriented gradients", University of Michigan, (2010).

[Van12a] Van Droogenbroeck, Marc, and Olivier Paquot. "Background subtraction: Experiments and improvements for ViBe", IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), p32-37, (2012).

[Wan10a]Wang, Weihong, Jian Zhang, and Chunhua Shen. "Improved human detection and classification in thermal images", IEEE 17th International Conference on Image Processing (ICIP), p2313-2316, (2010).

[Wan15a] Wang, Xin, Chen Ning, and Lizhong Xu. "Spatiotemporal saliency model for small moving object detection in infrared videos", Elsevier, Infrared Physics & Technology, 69, p.111-117, (2015).

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Segmentation of Fine Details in the CIELAB

Sergey V. Sai
Department of Computer Engineering
Pacific National University
Khabarovsk, Russia
sai1111@rambler.ru

Nikolay Yu. Sorokin
Department of Computer Engineering
Pacific National University
Khabarovsk, Russia
004040@pnu.edu.ru

Anatoly G. Shoberg
Department of Computer Engineering
Pacific National University
Khabarovsk, Russia
shoberg@rambler.ru

ABSTRACT

In the paper, we propose an algorithm of fine details segmentation in the CIELAB system considering the contrast sensitivity. For the search and segmentation algorithm we use a standard formula of the CIELAB color difference together with the weighting coefficients for the coordinates L^*, a^* and b^*. Experimental methods and results of the estimation of weighting coefficients are shown. Applications of the color model and segmentation algorithm are proposed.

Keywords

fine details segmentation, color space $L^*a^*b^*$, contrast sensitivity of human vision.

1. INTRODUCTION

The CIELAB color metric system is nowadays an international standard and is widely used for the estimation of color difference between original and distorted images. The differences are computed as

$$\Delta E = \sqrt{\left((L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2 \right)} \quad (1)$$

where (L^*_1, a^*_1, b^*_1) are the color coordinates of the first (original) image and (L^*_2, a^*_2, b^*_2) are the color coordinates of the second image. The value $\Delta E \approx 2.3$ approximately corresponds to the minimum perceptible color difference for the human eye [7, 10].

Color coordinates can be obtained using transformation of primary colors (*RGB*) into the color space (*XYZ*) and then using formulas [8]

$$L^* = 116\, f(Y/Y_n) - 16;$$
$$a^* = 500[f(X/X_n) - f(Y/Y_n)];$$
$$b^* = 200[f(Y/Y_n) - f(Z/Z_n)],$$

where

$$f(t) = \begin{cases} t^{1/3}, if\ t > 0.008856 \\ 7.787t + 16/116, if\ t \le 0.008856 \end{cases}$$

Values of X_n, Y_n and Z_n are the coordinates of the reference white color.

Equal color spaces CIELAB, CIELUV, CIEDE2000 and others are traditionally used for the estimation of the color rendering distortion of the big objects that have inside uniform color distribution. An advantage of such equal color spaces is that the estimation result weakly depends upon the object's color. S-CIELAB system [8] is proposed for the estimation of the color differences of complex objects. It uses spatial preprocessing of image and combines the traditional metric for color differences with the spatial properties of the human vision. It is achieved via preliminary object filtering before the pixel wise comparison. The difference metric of two images is expanded using the space-frequency adaptation and spatial localization blocks, as well as local and general contrast detection blocks [19].

The estimation of color difference using (1) fails with the decrease of object size because it does not take into account the decline of contrast sensitivity in brightness and chromaticity.

In [15, 16] proposed metric to estimate of color contrast fine details in the color coordinate system Wyszecki [18] considering the weighting coefficients of luminance and chrominance:

$$K_{WUV} = 3\sqrt{\left((\Delta W^*)^2 + (\Delta U^*)^2 + (\Delta V^*)^2 \right)} \quad (2)$$

where $\Delta W^* = \left(W_o^* - W_f^*\right)/W_{th}^*$ the normalized value of contrast on brightness index and $\Delta U^* = \left(U_o^* - U_f^*\right)/U_{th}^*$, $\Delta V^* = \left(V_o^* - V_f^*\right)/V_{th}^*$ on chromaticity index; W_o^*, U_o^* and V_o^* are the color coordinates of the fine detail from the test image;

W_f^*, U_f^* and V_f^* are the color coordinates of the background pixels; W_{th}^*, U_{th}^* and V_{th}^* are the weighting coefficients determined by the amount of the minimum perceptible color difference (MPCD).

Color coordinate values in brightness index (W^*) and the chromaticity indexes (U^*, V^*) calculated by the formulas [18]:

$$W^* = 25\ Y^{1/3} - 17;$$
$$U^* = 13 W^* (u - u_o);$$
$$V^* = 13 W^* (v - v_o),$$

where Y is the luminance, changed from 1 to 100; W^* is the brightness index; U^* and V^* are the chromaticity indices; u and v – are the chromaticity coordinates in Mac-Adam diagram [13]; u_o and v_o are the chromaticity coordinates of basic white color with u_o = 0.201 and v_o = 0.307.

Contrast sensitivity decreases with decreasing size of the detail, consequently, the weighting coefficients values in (2) will increase. In general, their values will depend on the brightness of the background, noise, the masking effect, conditions of observation and other [2]. For fine details with sizes not exceeding one pixel the threshold values are obtained experimentally. In particular [16], for fine details of the test table located on a grey background ($70 < W_f^* < 90$) threshold values are approximately $\Delta W_{th}^* \approx 6$ MPCD, $\Delta U_{th}^* \approx 72$ and $\Delta V_{th}^* \approx 80$ MPCD.

The formula (2), unlike the formula (1), does not estimate the color differences between two images, and determines the normalized color contrast of fine details within a single image. If the condition $K_{WUV} > 1$ is satisfied, then the fine detail is distinguished by an eye and the result weakly depends on the color of the fine detail.

Result of research [15] concludes that the application of $W^*U^*V^*$ color space has the following limitations:

1. Values of the weighting coefficients W_{th}^*, U_{th}^* and V_{th}^* greatly depend on the brightness of the background, which makes difficult to carry out the recognition and segmentation of fine details in photorealistic images with adequate results of visual perception.

2. $W^*U^*V^*$ system is now practically seldom applied because there are more effective systems CIELAB, CIEDE2000, S-CHIELAB and other for estimating the color differences [3, 8, 9, 12, 17, 19].

In this paper, we present our research results of the CIELAB system features in the tasks of fine details segmentation.

2. EXPERIMENTAL ESTIMATION OF THE WEIGHTING COEFFICIENTS

Color contrast of the fine details in CIELAB system is computed similar to (2):

$$K_{LAB} = \sqrt{\left((\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2\right)} \qquad (3)$$

where $\Delta L^* = \left(L_o^* - L_f^*\right)/L_{th}^*$, $\Delta a^* = \left(a_o^* - a_f^*\right)/a_{th}^*$ and $\Delta b^* = \left(b_o^* - b_f^*\right)/b_{th}^*$; L_o^*, a_o^* and b_o^* are the color coordinated of the fine detail; L_f^*, a_f^* and b_f^* are the color coordinated of the background; L_{th}^*, a_{th}^* and b_{th}^* are weighting coefficients determined by the amount of the MPCD on the lightness and chromaticity.

For the experimental estimation of the weighting coefficients we developed the test tables with fine details that are located on the uncolored background. Contrast of the fine details was specified separately for each coordinate L^*, a^* and b^*. For the image visualization the bmp format was used; pixel wise transformation from $L^*\,a^*\,b^*$ into *RGB* was carried out.

The following methodology was used during the experiment. Test tables with fixed fine details sizes (1 pixel, 2×2 pixels, 3×3 pixels and 4×4 pixels) were used for each experiment. Spatial position and the number of fine details were defined randomly. Background brightness has mean value L^*_f = 50.

Observer via the software interface changed the contrast for the coordinate L^* (a^* or b^*) for the selected table. This contrast was set from zero to some value when the fine details are being determined on the background by observer. At the end, observer fixed his subjective values L^*_{th} (a^*_{th} or b^*_{th}). Table 1 contains mean weighting coefficients values, determined by 20 observers. The reduction of contrast sensitivity characteristic depending on the size of details shows on Figure 1.

δ	L^*_{th}	a^*_{th}	b^*_{th}
> 4	1	3	4
4	1	10	15
3	2	16	20
2	3	20	30
1	6	40	55

Table 1. Dependence of weighting coefficients on the size of the fine detail

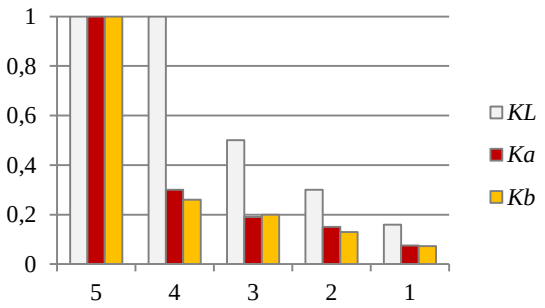


Figure 1. Dependence of the contrast sensitivity from the size of details, where $KL = 1/ L^*_{th}(\delta)$, $Ka = 3/ a^*_{th}(\delta)$ **and** $Ka = 4/ b^*_{th}(\delta)$

The experiments found that while changing the background brightness weighting coefficients vary insignificantly for fine details with size $\delta > 1$. However for the smallest details ($\delta = 1$) the values of the weighting coefficients on the coordinated $\pm a^*$ and $\pm b^*$ are on the border of their range. Question arises - should the color coordinates in the formula (3) be taken into account?

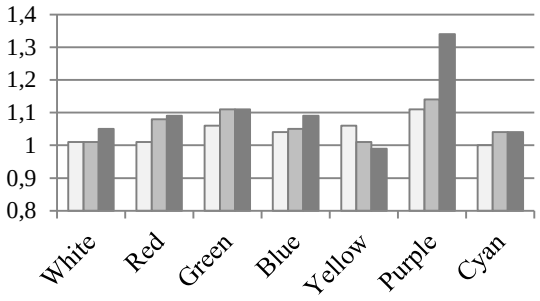
Here we should note that in the photorealistic images the change of the chromaticity coordinates for fine detail without change brightness is extremely rare. Therefore, we find it convenient to consider coordinates a^* and b^* in (3) with obtained weighting coefficients during the estimation of the smallest details color contrast, because they contribute to the final value of contrast.

To confirm the correctness of application of formula (3), series of experiments were carried out. We estimated the threshold values of the contrast of fine details for different colors. Table 2 contains the experimental results for the finest details ($\delta = 1$) for the primary and secondary colors. Dependences of the threshold contrast from the color and background brightness shows on Figure 2.

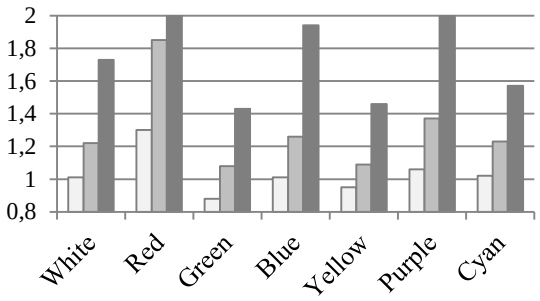
Software interface of the test table used the following functions: setting background brightness (Y_f) in the range 0…255; setting the colors of the fine details in the *RGB* system; changing the contrast of the fine details in *RGB* system relatively to the background; transformation from *RGB* to $L^* a^* b^*$ coordinates and computing the contrast using (3).

Color	*R*	*G*	*B*	L^*	a^*	b^*	K_{LAB}	K_{WUV}	Y_f
white	16	16	16	6.0	0	0	1.01	1.00	150
	15	15	15	6.1	0	0	1.01	1.22	100
	14	14	14	6.3	0	0	1.05	1.73	50
red	27	0	0	-5.1	21.0	8.5	1.01	1.30	150
	29	0	0	-5.2	24.7	10.9	1.08	1.85	100
	40	0	0	-3.3	34.8	22.7	1.09	4.59	50
green	0	24	0	4.6	-25.7	19.6	1.06	0.88	150
	0	23	0	5.0	-25.8	20.3	1.11	1.08	100
	0	20	0	5.2	-23.8	19.8	1.11	1.43	50
blue	0	0	18	-5.8	7.9	-19.0	1.04	1.01	150
	0	0	17	-58	8.4	-19.3	1.05	1.26	100
	0	0	16	-6.0	9.9	-20.3	1.09	1.94	50
yellow	18	18	0	5.9	-6.1	18.7	1.06	0.95	150
	16	16	0	5.7	-5.6	17.7	1.01	1.09	100
	14	14	0	5.6	-5.1	17.1	0.99	1.46	50
purple	30	0	30	-3.6	33.3	-22.3	1.11	1.06	150
	31	0	31	-3.3	36.1	-23.6	1.14	1.37	100
	48	0	48	1.2	48.3	-29.9	1.34	2.95	50
cyan	0	23	23	5.5	-15.2	-5.0	1.00	1.02	150
	0	22	22	5.8	-15.1	-4.8	1.04	1.23	100
	0	19	19	5.9	-13.4	-4.2	1.04	1.57	50

Table 2. Values of the color coordinates and threshold values of the fine detail



a) K_{LAB}



b) K_{WUV}

Figure 2. Dependence of the threshold contrast from the color and background brightness

For the given parameters of the background and contrast observer changed the contrast of the fine details until they do not become distinguishable. At this moment observer fixed contrast value K_{LAB}.

Contrast values K_{WUV} in the $W^*U^*V^*$ system computed using (2) are placed for the comparison.

From the analyses of the results we can conclude that the contrast threshold K_{LAB} insignificantly deviates from one (K_{LAB} = 1) with the change of the background color and brightness. Maximum deviation of the threshold (+0.34) is obtained for the purple fine details on the dark background (Y_f = 50).

Contrast threshold value K_{WUV} in the $W^*U^*V^*$ system significantly depends on the background brightness and has great scatter depending on color of the fine detail with the decrease of the background brightness. Maximum deviation of the threshold (+3.59) is obtained for the red fine details on the dark background (Y_f = 50).

Thus the threshold values of the CIELAB system more accurately correspond to the visual model and the application of the CIELAB system is preferable in the tasks fine details segmentation.

3. ALGORITHM OF FINE DETAILS SEGMENTATION

Description of known methods for search, recognition and object segmentation can be found in [7] and [14].

Fine details can be classified using the following properties: “dot object”, “thin line”, “texture element”. Search for dots and lines are simple algorithms presented in [7]. The most common search algorithm is an image processing using a sliding mask. As an example, for the 3×3 mask the processing is a linear combination of the mask coefficients with the brightness values of the elements, covered with this mask. Image distortions have high influence on the search and recognition of fine details. These distortions appear from the image digital compression and transfer over the noisy channels. The drawbacks of the known algorithms are: a) only brightness value is taken into account, and b) contrast sensitivity of the human vision [2, 17], is not considered at all.

We propose an algorithm of fine details segmentation in images based on the contrast measurement in the uniform color space CIELAB. Consider fine details (δ = 1) segmentation algorithm step by step.

1. Make pixel wise transfer from *RGB* into $L^*\,a^*\,b^*$.

Figure 3. Test image “Man”

2. Select first micro-block using the mask (2×2 pixels) and compute its contrast (3), where $\Delta L^* = \left(L_i^* - L_j^*\right) / L_{th}^*$, $\Delta a^* = \left(a_i^* - a_j^*\right) / a_{th}^*$ and $\Delta b^* = \left(b_i^* - b_j^*\right) / b_{th}^*$; L^*_{th}, a^*_{th} and b^*_{th} are the values of the weighting coefficients from the Table 1 for the fine details with size equal to one pixel; *i* is the pixel number with maximum color coordinates values (L^*_{max}, a^*_{max}, b^*_{max}) and *j* is the pixel number with minimum color coordinates values (L^*_{min}, a^*_{min}, b^*_{min}).

3. Check the condition

$$K_{LAB} > Th, \quad (4)$$

where Th is a contrast threshold when the neighbor pixels in the micro-block are distinguished by an eye.

4. If the condition (5) is fulfilled then the decision on membership of the fine details in the micro-block is taken. The micro-block is marked by a marker $m_1(k, i, j) = 1$, where k – micro-block number with spatial coordinates (i, j).

5. If the condition (5) is not fulfilled then the micro-block does not have any fine details that are distinguished by an eye.

6. Slide the mask one pixel forward to the next micro-block and repeat steps 2-5.

Figure 4. Result of segmentation for $\delta = 1$
(micro-blocks 2×2; Th = 1.5)

After segmentation of the finest details we move to the segmentation algorithm for the 2×2 pixels (δ = 2).

1. Select the first block with the mask 4×4 pixels and divide it into four micro-blocks 2×2 pixels.

2. Check the micro-blocks with marker m_1. If there is from zero to two such micro-blocks then we analyze this block: compute mean value of the color coordinates $(L^* a^* b^*)$ for the micro-blocks with $m_1 = 1$ and estimate the contrast using (3), where L^*_{th}, a^*_{th} and b^*_{th} – values of the weighting coefficients from the Table 1 for the fine details with size δ = 2; i – micro-block number with maximum color coordinates values and j – micro-block number with minimum color coordinates values.

3. If number of micro-blocks with marker m_1 is greater than two then slide the mask forward with step equal to two pixels and process the next block.

4. If the condition (4) is fulfilled then mark the micro-blocks using marker $m_2(k, i, j) = 2$ and make segmentation of these micro-blocks.

5. If the condition (4) is not fulfilled then slide the mask forward with step equal to two pixels and process the next block.

Figure 5. Result of segmentation for $\delta = 2$
(micro-blocks 4×4; Th = 1.5)

Segmentation of the fine details with size of 3×3 and 4×4 pixels can be carried out in similar way.

Figure 6. Result of segmentation for $\delta = 4$
(micro-blocks 8×8; Th = 1.5)

Selection of the threshold value Th in formula (4) depends upon the next factors. It follows from the experimental data (Table 2) that when $K_{LAB} \approx 1$ the fine details (dot objects on the uniform uncolored background) from the test table begin to differ. Contrast of the block depends on the masking effect of the neighbor blocks for the photorealistic images. Thus, we propose to set the threshold value to Th = 1.5.

Figure 7 presents test image "Man" and segmented fragments with fine details for Th = 1 and Th = 2.

a) $\delta = 1$, $Th = 1$ b) $\delta = 2$, $Th = 1$

c) $\delta = 1$, $Th = 2$ d) $\delta = 2$, $Th = 2$

Figure 7. Results of test image segmentation for different threshold values

Thus, the proposed algorithm can segment the fragments of photorealistic images with fine details of a given size considering the contrast sensitivity of the human vision. Additionally, the algorithm allows estimation of the level of fine details (*FDL*) [5, 6] as

$$FDL = 4 \cdot N_{m1} / (W \cdot H),$$

where N_{m1} – number of segmented micro-blocks with 2×2 pixels size; W and H – width and height of the image in pixels.

Experimental results of segmentation algorithm application to the high quality digital images show that it produces adequate output.

If the image is distorted with noises then having low signal to noise ratio (PSNR) leads to identifying the noise components as fine details and the *FDL* value will grow. Artifacts of JPEG or JPEG 2000 compression algorithms with high compression level (low quality) lead to decrease of the *FDL* value due to the blurring of the fine details.

Experimentally we conclude that the *FDL* value changes slightly with PSNR > 41 dB and/or JPEG compression with high quality settings.

4. CONCLUSION

Here we consider an application method of the proposed algorithm in the fine details transfer quality assessment system [1, 4, 9, 11, 17, 19]. For the preliminary quality change compare the *FDL* values of the original (reference) and transformed images after the compression, filtering and other operations. Decrease of the *FDL* value means the definition reduction of the original image.

For qualitative analysis, we offer the following procedure.

1. Apply segmentation algorithm to the original image and receive 4 spatial masks for the regions with fine details: 1 pixel (m_1), 2×2 pixels (m_2), 4×4 pixels (m_4) and more than 4 pixels (m_5).

2. For each selected region with fine details (m_1, m_2, m_4) compute mean deviation of the contrast between original and transformed images (ΔKm_1, ΔKm_2, ΔKm_4) in accordance with the weighting coefficients (Table 1) and formula (3).

3. For the big details and background pixels (m_5) compute mean deviation (ΔKm_5) of the color coordinates in CIELAB system in accordance with the formula (1).

Thus we get the four parameters of color coordinates deviation of original and transformed image. These parameters must be compared with given threshold for visual estimation. We propose to select the threshold value less than 5…10% from the computed values in steps 2 and 3.

Ultimately, the proposed method makes it possible to estimate the reduction of the visual definition by following simple criterion: if the distortion exceeds the threshold, the definition is low; if distortion does not exceed the threshold then the definition corresponds to a high quality.

For dependencies of the visually quality on the selected rating scale from to the parameters (ΔKm_1, ΔKm_2, ΔKm_4 and ΔKm_5) in compression systems (JPEG, JPEG2000, etc.) more research is needed.

5. REFERENCES

[1] Bovik, A. and Mittal, A. No-Reference Image Quality Assessment in the Spatial Domain. IEEE Transactions on Image Processing. 21(12), pp. 4695 – 4708, 2012.

[2] Barten, P.G.J. Contrast sensitivity of the human eye and its effects on image quality. Knegsel: HV Press, 1999.

[3] Sharma, G., Wu, W. and Dadal, E. The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data and Mathematical Observations. Color Research and Application. – 2004, Feb 09.

[4] Comaniciu, D. and P. Meer, P. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(5): pp. 603-619, May 2002.

[5] Dosselmann, R. and Xue Dong Yang. No-Reference Image Quality Assessment using Level-of-Detail. Technical Report CS 2011-2, May 2011.

[6] Dosselmann, R. and Xue Dong Yang. Mean shift particle-based texture granularity. Proc. 2008 IEEE Int. Instrumentation and Measurement Technology Conf., pp. 101-106, May 2008.

[7] Gonzalez, R.S. and Woods, R.E. Digital Image Processing. Prentice Hall. New Jersey, 2002.

[8] Fairchild M.D., Color Appearance Models, John Wiley and Sons., 2005.

[9] Fairchild, M.D. and Johnson, G. Meet iCAM: A next-generation color appearance model. Proceedings of the tenth Color Imaging Conference (IS&T/SID, Scottsdale, Arizona, 2002), pp. 33-38, 2002.

[10] Judd, D.B. Color in business, science and industry. John Wiley & Sons, 1975.

[11] Lin, W. and C-C Jay Kuo. Perceptual Visual Quality Metrics. Visual Communication and Image Representation. 22(4), pp. 297-312, 2011.

[12] Moroney, N., Fairchild, M.D., Hunt, R.W.G., Changjun Li, M. Ronnier Luo and Todd Newman. The CIECAM02 Color Appearance Model. Proceedings of the tenth Color Imaging Conference (IS&T/SID, Scottsdale, Arizona, 2002), pp. 23-27, 2002.

[13] MacAdam, D.L. Visual Sensitivies to Color Differences in Daylight. J. Opt. Soc. Am., Vol. 32. 5, pp. 247-274, 1942.

[14] Pratt, W.K. Digital Image Processing. Wiley, 2001.

[15] Sai, S.V. and Sorokin, N.Yu. Search Algorithm and the Distortion Analysis of Fine Details of Real Images. Pattern Recognition and Image Analysis, 19(2), pp. 257-261, 2009.

[16] Sai, S.V. Methods of the Definition Analysis of Fine Details of Images. Chapter in the book: Vision Systems: Applications, G. Obinata and A. Dutta (eds.), I-Tech Education and Publishing, pp. 279-296, 2007.

[17] Watson, A.B., James Hu, and John F McGowan. Digital video quality metric based on human vision. Journal of Electronic Imaging, 10(1), pp. 20-29, 2001.

[18] Wyszecki, G. Uniform Color Scales: CIE 1964 U*V*W* Conversion of OSA Committee Selection. JOSA. 65, pp. 456-460, 1975.

[19] Zhang, X., Silverstein, D., Farrell, J., and Wandell, B. Color image quality metric S-CIELAB and its application on halftone texture visibility. [Compcon '97. Proceedings, IEEE], pp. 44–48, 1997.

A Haptic Simulator for Studying Rest-To-Rest Reaching Movements in Dynamic Environments

Igor Goncharenko
3D Incorporated
2-3-8 Shin-Yokohama
222-0033, Yokohama, Japan
igor@ddd.co.jp

Mikhail Svinin
Kyushu University
744 Motooka, Nishi-ku,
819-0395, Fukuoka, Japan
svinin@mech.kyushu-u.ac.jp

ABSTRACT

We present a haptic simulation system with interchangeable physical constraints for studying skillful human movements. The unified haptic interface easily links different physical models with 2D and 3D static spatial constraints and graphical content related to the models. The system was tested on a variety of reaching tasks performed by human subjects. In the experiments, we analyzed motions based on data recorded by a history unit with a frequency of 100Hz. Theoretical and experimental kinematic profiles were compared for several cases of basic reaching rest-to- rest tasks, namely, line-constrained movement during transport of flexible object and parallel flexible object. Experimental patterns exhibit a good agreement with theoretical optimal control models based on jerk and force-change minimization criteria.

Keywords

Haptic interface, rest-to-rest movement, dynamic environment, optimality

1. INTRODUCTION

Numerous haptic applications have demonstrated subjectively realistic modeling of kinesthetic and tactile sensations of virtual reality (VR) object properties as such as mass, inertia, shape, viscosity friction, vibration, stiffness, and roughness. Many of these applications deal with *constrained* human movements, but little is known about movement formation in the constrained real and virtual environments (VE). In addition to practical (e.g., VR rehabilitation [Bur03]) and entertainment applications, simulators can be used for basic research in computational neuroscience (CN) studying movement trajectory formation and invariant features of movements.

Consider, for instance, point-to-point and rest-to-rest reaching tasks, typical in VR rehabilitation [Pir03]. If a static three-dimensional (3D) surface- or curve-based constraint, e.g., an ellipsoid, or a circle, is used in a haptic system as a VR constraint, the user's hand trajectories follow the specified 3D curve or lie on the surface. In CN research, unconstrained reaching exhibits invariant features as such as low curvature and bell-shaped velocity profiles. Invariant features change when movements are constrained by curves or surfaces. We do not, however, know how specifically they change or how change is related to constraint geometry and human visual feedback.

To clarify the problem of constraint hand movement formation in rest-to-rest reaching tasks, this paper presents an analysis of human movements in manipulation of flexible objects. This analysis is based on experiments completed with a haptic simulator.

Related works

Developing mathematical models and optimality criteria for predicting human movements constrained by the environment remains an open research area in CN. Some criteria [Fla85, Fla03, Uno89, Svi04a, Din04, Lei12, Mor95] are given in **Table 1**. In optimization approaches, the trajectory of the human arm is found by minimizing, over movement time T, integral performance index J subject to boundary conditions imposed on start and end points. In **Table 1**, x is the hand contact point vector, f is the force applied to the end point, τ is the vector of arm joint torques, and x_{cm} is the center of mass (CoM) of the system "hand-object". (Note, that superscript (5) in **Eq. (3)** means the 5th derivative.)

Minimum jerk is commonly accepted criterion in CN. However, numerous experiments of hand movement capturing in haptic environments were done by using simple "one mass – one spring" dynamic object model. For such a simple model, hand and object movement trajectories predicted by

different criteria may be very similar. In some cases [Lei12], the minimum hand jerk criterion is rejected as not applicable. To correctly discriminate the criteria, hand movement should be studied by interaction with complex dynamic environment , like multiple mass-spring objects. Right selection of optimality criterion is important for such areas as robotics, computer animation, CN, biological cybernetics.

The advent of haptic technology is making it possible to confirm or disapprove movement prediction criteria because potentially any type of VR constraint may be implemented in systems. Typically, haptic interaction is simulated using collision detection, often accelerated by GPU-based calculations (e.g., [Kal14, Vei09, Wel11]). As human rest-to-rest movements are smooth, we do not use collision detection, but utilize smooth analytical constraints (with smooth derivatives), which are parametric curves and surfaces in 3D space. In this case, object - constrained dynamic integration is very fast and does not require any parallelism, or separate threading of simulation. To do so, we built a haptic visualization environment. During design, we first required that constraints should be easily interchangeable and linked to the physical simulator core to study human arm movements in different constrained VEs.

Criterion name	Performance index	Ref
Minimum jerk	$J = \int_0^T \dddot{x}^T \dddot{x}\, dt$	(1)
Minimum joint torque change	$J = \int_0^T \dot{\tau}^T \dot{\tau}\, dt$	(2)
Minimum crackle	$J = \int_0^T x^{(5)T} x^{(5)}\, dt$	(3)
Minimum hand force	$J = \int_0^T f^T f\, dt$	(4)
Minimum hand force change	$J = \int_0^T \dot{f}^T \dot{f}\, dt$	(5)
Minimum CoM acceleration	$J = \int_0^T \ddot{x}_{cm}^2\, dt$	(6)

Table 1. Optimality criteria for movement prediction.

There are three novel aspects considered in this paper:

- Usage of changeable analytical constraints in haptic simulators instead of collision detection;
- Modeling of dynamic environment as flexible objects, namely, multiple mass-springs connected in sequential or parallel manner and following the spatial constraints;
- Usage of the proposed simulator in experiments to approve human hand rest-to-rest motion planning strategy in accordance with the minimum jerk / minimum hand force change optimization criteria.

Section 2 discusses the distributed architecture connected single point force devices via networks to study cooperative and collaborative arm movements. Section 3 describes the use of changeable spatial constraints in the physical-based simulation module. Note, that the method of constraint generation calculates not only dynamic coefficients, but also coordinates of curve/surface in 3D. This can be instructive for graphics community specialized in parametric curve/surface modeling and rendering. Sections 4-5 demonstrate controlling flexible VR objects. These sections compare collected haptic experimental data with theoretical optimality criteria. Section 6 presents conclusions.

2. HAPTIC SYSTEM DESIGN

We built our system (**Fig.1**) based on two dual-CPU PCs (server and client), interconnected via Ethernet, and each equipped with its own point force device.

Industry-standard PHANToM devices originally developed at MIT [Sal97] are suitable for studying constrained human movement. In our case, SensAble/Geomagic PHANToM 1.5/6.0, PHANToM High Force, and Omni manipulators controlled through Open Haptic Toolkit [Geo] were used. Critical loops in the overall control scheme include haptic rendering, graphical rendering, and a simulation loop. We focused on the efficiency of haptic and simulation loops to achieve real-time capabilities and robust realistic interaction via point-force devices in constrained VEs.

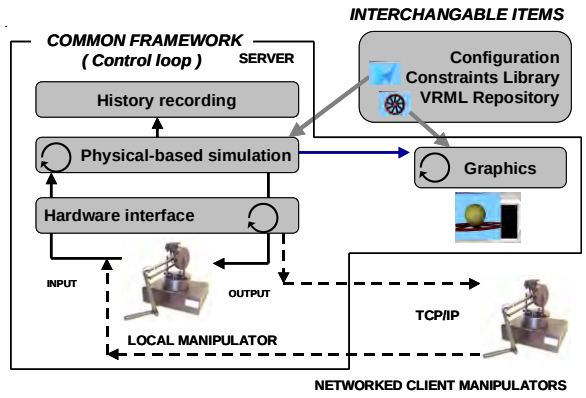


Figure 1. System architecture.

To support haptic simulator cloning, new dynamic models are reduced to the following standard *N*

ordinary differential equations (ODE) with *M* time-dependent parameters:

$$dy_i / dt = f_i(y_{1,}...y_{N,}c_1(t),...,c_M(t)) \quad , (7)$$

where parameters $c_i(t)$ ($1 \le i \le M$) are control functions. Different constraints f_i are linked to the physical simulator from the external constraint library. During simulation, system (7) is integrated by the Runge-Kutta 4th-order method for the time step 0.001s, defined by the constant haptic cycle of PHANToM devices. Typically, controls $c_i(t)$ are feedback force and moment components. To calculate feedback, we introduced a fixed point (FP) for Hooke's and spring-damper models [Bur03]. At the start of haptic interaction, FP is coincided with haptic interface point (HIP, or proxy), and it is considered as rigidly bound to the VR body during simulation. Distance $\Delta r(t)$ between current HIP and FP defines force $F(t)$ applied to the VR body:

$$F(t) = k_h \Delta r(t) + b_h \Delta \frac{dr(t)}{dt} \ , \qquad (8)$$

where k_h, b_h are coefficients of the spring-damper model.

Force components of (8) are used in **Eq.(7)**, and generated haptic feedback force is just opposite to the force given in **Eq.(8)**. In constrained VEs, this models human movements such as hook-and-carry and catch-and-move.

The history unit records all simulation data: time-dependent parameters, feedbacks, object and hand positions, velocities and accelerations. Recording is performed at a frequency of 100Hz, sufficient to analyze basic human motions with the average reaction time above 200ms. The system required certain flexibility in different constraints, attained by developing two additional parts – a configuration repository and a constraints library. The configuration module defines initial dynamics model values, graphical scene representation (references to VRML scenes), and static parameters such as mass, inertia, and viscosity friction. VRML objects are completely independent and are replaced in the configuration repository.

Constraint types and shapes are defined analytically in the constraints library. For parametric surface- and curve-constraints, we developed a partially semiautomatic procedure to generate functions f_i (7). A general library written in Mathematica [Wol03] is processed, with each surface/curve type in this library defined in a simple analytical form by surface/curve radius-vector components. Partial derivatives of the radius-vector and necessary dynamics coefficients (Section 3) are calculated as analytical expressions, which are exported in C-code by Mathematica, compiled, and linked to an ODE solver to be used in the simulation loop.

Several haptic devices (clients) connected to the server via Ethernet had to be supported to study two-hand cooperative and multi user collaborative movements (e.g., [Gon04]). **Fig.1** demonstrates the simplest client-server configuration. In this research, only one manipulator is used.

As the number of ODEs is very small for curve/surface constraints and right parts of equations (7) can be expressed in analytical form, one time step integration of ODEs to calculate dynamic environment (in case of mass-spring connections, positions of the centers of masses) is negligible in comparison with the haptic cycle (0.001s). That is, physical simulation (**Fig**.1) can be implemented directly in the haptic thread. Therefore, a simulator similar to the described one can be implemented as two-thread CPU-based application. In this case, the graphical rendering thread gets positions of masses calculated in the haptic thread. Functionality of the haptic thread is straightforward: the thread receives HIP position, calculates object driven force (and, haptic force as opposite to the driven force) by (8), calculates right parts of equations (7), performs integration by the Runge-Kutta method for the time step 0.001s (correspondent to the haptic cycle) to get positions of masses, and, finally, applies haptic forces to the haptic device. In the above calculation scheme, subject's rest-to-rest movement trials are realized as follows. When dynamic system is at the start position, driven/haptic forces, masses' accelerations and velocities are zeroed, that is the dynamic system is at rest. A movement trial starts by application of non-zero forces (8) and continuous integration is performed. When the system reaches the target position (with some tolerances on velocities/accelerations), the forces are zeroed again. When a signal to proceed with the next trial appears, the system is placed to the rest start position, and so on.

3. MODELING OF CONSTRAINTS

Different 2D and 3D constraints are derived, reduced to form **Eq.(7)** and linked to the simulator. Movements are assumed applied to VR objects via a single haptic interface. Realistic rigid body (or, flexible object) sensations are achieved when stiffness coefficients (k_h in **Eq.(8)**) for feedback exceed 500N/m. For such values, force damping and clamping may be required for fast movements because PHANToM's maximum apparatus load is 12N (37N for PHANToM High Force). During the

course of our experiments, we configured the system to avoid exceeding of the force limits.

Consider a point of mass m in viscosity field λ. Assume that the point is loaded by external force $f = (f_x, f_y, f_z)^T$. Unconstrained dynamics are defined by

$$m\ddot{r} + \lambda\dot{r} = f \quad , \tag{9}$$

where $r = (x, y, z)^T$ is the radius-vector of the point. Assume now that the point is constrained by a 3D curve. The constraint curve is given by

$$r(\varphi) = \big(x(\varphi), y(\varphi), z(\varphi)\big)^T \tag{10}$$

for $\varphi \in [0, 2\pi]$.

Differentiating **Eq.(10)** and defining $\omega \equiv \dot{\varphi}$, the physical model of curve-restricted motions is then described by the following two first-order ODEs:

$$\dot{\varphi} = \omega,\ M(\varphi)\dot{\omega} + L(\varphi)\omega + V(\varphi)\omega^2 = r_\varphi^T f \ , \tag{11}$$

where

$$M = m\big(r_\varphi^T r_\varphi\big), L = \lambda\big(r_\varphi^T r_\varphi\big), V = m\big(r_\varphi^T r_{\varphi\varphi}\big),$$
$$r_\varphi \triangleq \frac{\partial r}{\partial \varphi} \ , r_{\varphi\varphi} \triangleq \frac{\partial^2 r}{\partial \varphi^2} \ .$$

Dynamic equations (11) now match the form (7) and are used for the simulator. Such model parameters as mass of point m and viscosity coefficient λ are defined in the configuration repository.

By analogy, equations for surface constraints are derived using **Eq.(9)** and assuming that movements are constrained by the (u,v)-parametric surface:

$$r(u, v) = \big(x(u, v), y(u, v), z(u, v)\big)^T \quad . \tag{12}$$

After finding derivatives of r, equations of the constrained system in coordinates (u,v) are:

$$mA(u, v)\begin{pmatrix}\ddot{u}\\ \ddot{v}\end{pmatrix} + \lambda B(u, v)\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} + mC(u, v, \dot{u}, \dot{v}) = Q(u, v, f) \quad . \tag{13}$$

Elements of matrices A, B, and vectors C, Q depend on partial derivatives of radius-vector r by u and v and are found analytically. Equations (13) are rewritten as four first-order ODEs to fit form (7). The above analytical calculations are done automatically, and only basic curve/surface expressions (10) and (12) are needed to be defined. Note, that parameters u,v, φ are non-dimensional.

As an example, consider step by step the automatic constraint generation for task (11). Initially, two standard wrapping C-code patterns without calculation expressions (heads or tails) are created: the first one is used to interface dynamic constraints with a module solving ODEs (7), while the second one links curve coordinate calculations with a 3D graphical rendering module. The following unified script in Mathematica is then run:

```
(* VARIABLE PART: Curve definition *)
   Curve3d[a_,b_][fi_]:={0,aCos[fi] , bSin[fi]};
(* COMMON PART *)
(* Input forces and radius-vector*)
   f={fx,fy,fz} ;  r=Curve3d[a,b][fi];
(* Output coordinates of this 3D-curve *)
   x=r[[1]]; y=r[[2]]; z=r[[3]];
(* Derivatives *)
   rfi=Simplify[D[r,fi]]; dxdfi=rfi[[1]]; dydfi=rfi[[2]];
   dzdfi=rfi[[3]]; rfifi=Simplify[D[rfi,fi]];
(*  Coefficients of the dynamic equations  *)
    M=m Simplify[rfi.rfi];
    L=lambda Simplify[rfi.rfi];
    V=m Simplify[rfi.rfifi]; Q=Simplify[rfi.f];
(*Generate C- code  *)
    StringForm
    [
    "/*---------- Curve coordinates --------*/\n
    x = ``;\n   y = ``;\n   z = ``;\n
    /*---------- Dynamic parameters -------*/\n
    M = ``;\n  L = ``;\n   V = ``;\n  Q = ``;\n
    /*---------- END OF C-CODES -----------*/\n",
    CForm[x],   CForm[y],   CForm[z], CForm[M],
    CForm[L],   CForm[V], CForm[Q]
    ]
```

The script calculates analytically coordinates on the curve in 3D and dynamic parameters *M*, *L*, *V* given in formula (11). Operator "D" calculates partial derivatives, and operator "Simplify" fulfills analytical simplification (e.g., trigonometry, or algebra simplification). Operator "CForm" generates C-code to be used in haptic application. In the script, parts emphasized by bold font are variable. In this case, it represents a 3D ellipse. The expressions generated are automatically post-processed for further trigonometry optimization, merged with the wrapping patterns, compiled in batch mode, and added to the current constraint library.

Only simple analytical expressions, similar to the above one-line 3D ellipse definition, must be stored in and added to a source constraint library. At present, more than 30 such definitions are used for cloning haptic simulators with spatial constraints. GUIs with some of these constraints (epitrochoid, monkey saddle, plane, torus in 3D) are shown in

Fig.2. In the figure, small spheres represent start and stop positions for the driven object (larger sphere).

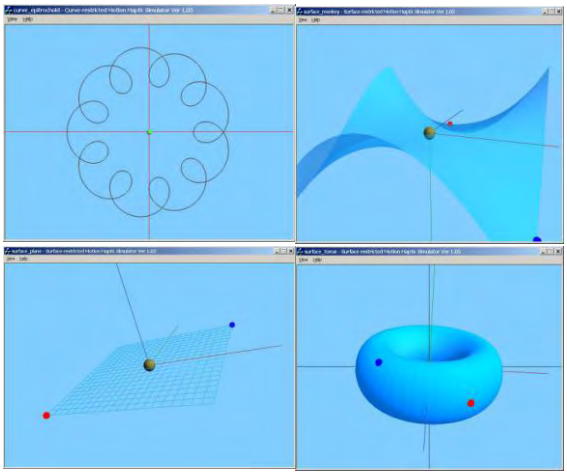


Figure 2. Curve- and surface-based constraints.

While conducting the requested point-to-point movement of the VR body, users can sense inertia of the driven object, the viscosity field, and the shape and curvature of the constraint surface. To control movement at arbitrary locations on surfaces, graphical rendering is done semi transparently or in a wire-frame. Haptic feedback is calculated using FP method (8).

Constraint changeability becomes very useful when a movement prediction criterion must be checked for a variety of constraint types. Below, we compare theoretical results based on different criteria (**Table 1**) with experimental data collected via the haptic system. For the experiments described in next sections the system was initially configured by selecting line constraint in 3D

$$r(\varphi) = \left(x_B + \frac{\varphi}{2\pi}(x_E - x_B)\right)^T, \qquad (14)$$

where $x_B = -0.1$ $x_E = 0.1$. These constants allow us to simulate constrained movement in haptic environment along horizontal line in the range of 20cm.

4. MOVEMENT OF FLEXIBLE OBJECTS

In addition to geometrically constrained movements, we also considered point-to-point rest-to-rest constrained movement for flexible objects, which may require long training and good skills from the system users. In [Din04], the simplest flexible VR system consists of a single mass, which humans can interact with through a haptic interface with a stiffness of 120N/m. We implemented multi mass system modeling to check hand movement optimality criteria. The flexible object (**Fig.3**) is modeled by several masses connected by damping springs, and external haptic force f_h is applied to the driving mass (right large sphere).

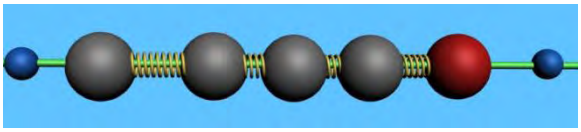


Figure 3. Flexible object model.

Masses move along a 3D curve and penetrate each other virtually, yielding very complex oscillation. Equations describing the flexible VR object for arbitrary 3D curve-constraint are derived, using formulas (9) for the case of N masses, so we have 2xN first-order ODEs:

$$\frac{d\varphi_i}{dt} = \omega_i, \quad 1 \le i \le N,$$
$$\frac{d\omega_i}{dt} = [r'^T_{\varphi_i}(\varphi_i) f_i - L(\varphi_i)\omega_i - V(m_i, \varphi_i)\omega_i^2] / M(m_i, \varphi_i),$$

where

$$f_i = k_{i-1}(r_{i-1} - r_i) + k_i(r_{i+1} - r_i) + b_{i-1}(\dot{r}_{i-1} - \dot{r}_i) + b_i(\dot{r}_{i+1} - \dot{r}_i) + g, \quad 1 < i < N$$
$$f_1 = k_1(r_2 - r_1) + b_1(\dot{r}_2 - \dot{r}_1) + f_h + g,$$
$$f_N = k_{N-1}(r_{N-1} - r_N) + b_{N-1}(\dot{r}_{N-1} - \dot{r}_N) + g.$$

g is the gravity acceleration, $f_h$is the external haptic force, $\dot{r} = \frac{\partial r}{\partial \varphi}\dot{\varphi}$, k_i, b_i are spring stiffness and damping coefficients. As derivatives of r found from (14) are constant, after setting λ=0 the above dynamic equations have classic Newton's law form. Prior to experiments, the system was configured to be constrained by a straight line. Gravity and line viscosity were set to zero. To be compatible with experimental results published by other researchers, all damping coefficients were also set to zero. Five equal 0.6kg masses are connected by springs (**Fig.3**) and all spring stiffness coefficients are equal to 600N/m. The PHANToM stiffness coefficient ($k_h$in (8)) is also 600N/m.

The reaching task was formulated for experimenters so that, initially, all masses are at rest and coincide at the initial point (small left sphere in **Fig.3**). Users were instructed to move the 5-mass system to the target point (small right sphere) during designated time T. All masses should finally be at rest and coincide at the target point. The travel distance was set to 0.2m. Tolerances were introduced to count successful reaching trials: position deviation, speed, and time tolerances $\Delta x, \Delta v, \Delta T$; and all masses must

obey the tolerances. When a reaching task is successful, haptic interaction is stopped and an audio signal prompts the user to proceed with the next trial.

Fig.4 schematically illustrates ergonomics of subjects during the experiments.

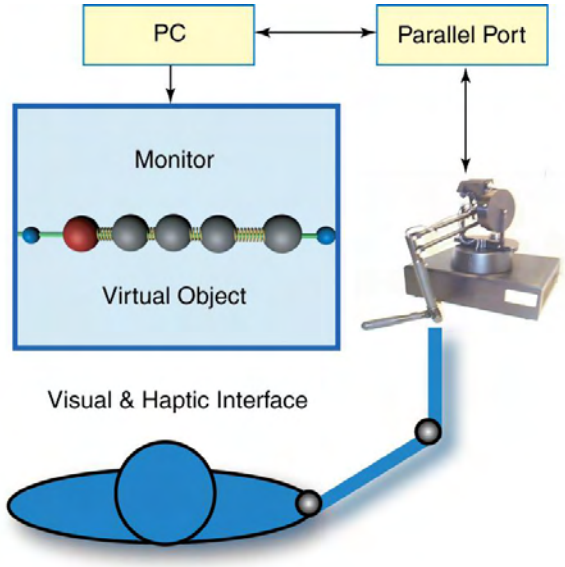


Figure 4. Experimental environment for movement of flexible object.

One subject conducted preliminary experiments and defined three tolerance sets at the subject's own pace for slow, moderate, and fast movements to make experimental results statistically representative. Procedure for defining the tolerance set for *moderate* movements is described below.

Reaching movements under consideration are quite unusual from what we experience in daily life movements, and an experiment – similar to [Din04] – was conducted in two days. On the first day, the subject was familiarized with the experimental setup, learned the unusual dynamic environment, and performed trial movements. Initially, the subject was asked to complete reaching during $T = 1.00 \pm 0.5\text{s}$ within tolerance windows $\Delta x = \pm 0.006\text{m}$, $\Delta v = \pm 0.006\text{m/s}$. It turned out that the learning of successful movements constituted only 5% of 100 trials. The low learning rate is attributed to the relatively narrow time, position, and velocity windows.

To facilitate learning, two windows were set as follows: $\Delta x = \pm 0.012\text{m}, \Delta v = \pm 0.012\text{m/s}$. On the 1st day the subject made 2 series of 100 trials, with overall success rate of about 10%. On the 2nd day the subject made 2 series of 100 trials, with overall success rate increasing to 17%. The average movement time become 1.35s (maximal 1.49s, minimal 1.13s, and standard deviation from average 0.09s). Similarly, slow and fast movement tolerances were as following :

Slow: $T = 2.26 \pm 0.5\text{s}$,
$\Delta x = \pm 0.006\text{m}, \Delta v = \pm 0.006\text{m/s}$;
Fast: $T = 0.68 \pm 0.5\text{s}$,
$\Delta x = \pm 0.012\text{m}, \Delta v = \pm 0.024\text{m/s}$.

Five subjects (4 men and one woman) participated in experiments based on the same scheme:

- All three tolerance sets were fixed as described above;
- On the first day, subjects made 100 preliminary trials for each slow, moderate, and fast movement task;
- On the second day, subjects made 100 additional trials for each slow, moderate, and fast movement task.

All experimental sets for all subjects demonstrated very similar results in favor of the minimum jerk criterion (1). Here, only the results for reaching time $T = 1.35\text{s}$ for one subject are shown.

Experimental velocity profiles, time-scaled to the average, are shown in **Figs.5** and **6** by thin lines. Hand and object velocity profiles, predicted by criteria (1) and (3) for constraints (12), are shown by thick solid and thick dashed lines, respectively. Note that the last fifth mass's velocity is given as "object velocity."

Experimental data favors the minimum hand jerk criterion. Experiments with one mass of 3kg and PHANToM's stiffness equal to 120N/m were also conducted to check results reported in [Din04]. For this configuration, predicted velocity profiles are very close in magnitude and shape for both (1) and (3) criteria. In [Svi04b, Svi06] it was proved that the minimum crackle criterion does not converge to criteria (1) when stiffness is increased. When number of masses N is increased, the criterion (3) gives unconstrained velocity profiles, asymptotically approaching the Dirac delta function.

All subjects showed progress in motor training from Day 1 to Day 2 (D1, D2 in **Table 2**). Note that subject S1 established tolerance first for moderate, then for fast, then for slow movement, i.e., participating in 6 experiments. Subject S2 volunteered on two additional days, making 2 sets of experiments daily for each of the movements. In the table, S, M, and F mean slow, moderate and fast movements.

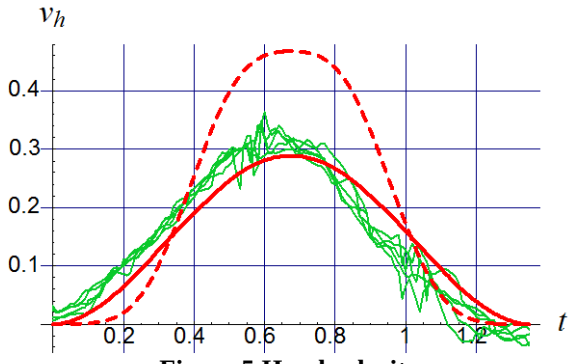


Figure 5 Hand velocity

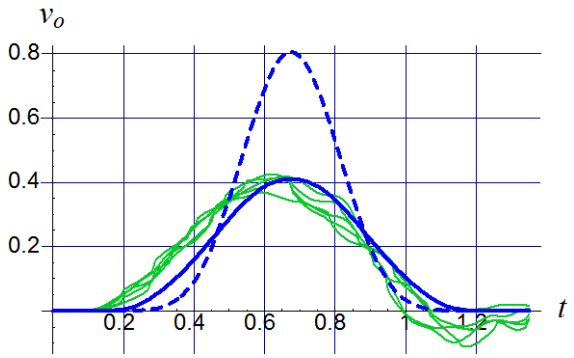


Figure 6 Object velocity

Subj	S D1	S D2	M D1	M D2	F D1	F D2
S1	64	76	10	17	14	23
S2	17	36	31	44	17	31
S3	40	73	35	47	19	28
S4	46	93	41	82	20	51
S5	32	55	25	48	17	27

Table 2 Progress in motor learning (success, %)

5. PARALLEL FLEXIBLE OBJECTS

Studying of flexible objects transport in haptic environments was carried by several researches. However, the majority of experimental works deals with only one mass virtually "connected" to human hand via the haptic proxy. The advantage of our system is that it can simulate highly dynamic environment with several masses, connected by springs. After some configurations of the system, experimental data can be collected to make choice in favor of one of the criteria (1)-(6). Not only bell-shape velocity profiles can be observed; for instance, two- and three-phase profiles were observed, that match well to theoretical profiles [Svi06, Gon10].

Recently, a novel model, named as the minimum acceleration of the center of mass (6), has been proposed and tested against experimental data for a single mass flexible object [Lei12]. In the theoretical justification of this model it is argued that neither the minimum hand jerk model (1) nor its dynamic counter- part, the minimum hand force change model (5), are applicable to modeling of reaching movements with parallel flexible objects.

Contrary to the above statement, we demonstrated that the invariant features of hand trajectories in the manipulation of parallel flexible objects can be well captured by the minimum jerk hand model, and theoretical solution for 2-mass-hand system was found [Svi16].

From the standpoint of haptic dynamic simulation, change of haptic force is needed (spring model without damping):

$$F(t) = k_1(x_1 - x_h) + k_1(x_2 - x_h) \quad .$$

And, the motion equations are:

$$m_1\ddot{x_1} + k_1(x_1 - x_h) = 0,$$
$$m_2\ddot{x_2} + k_2(x_2 - x_h) = 0,$$

where m_1, m_2, k_1, k_2 , x_1, x_2 are masses, spring stiffness, and coordinates of first and second mass, and x_h is the hand coordinate (HIP). The above expressions were used to build a new constraint, which was added to the haptic simulator's solver.

In (**Fig.7**) light smaller sphere center is the human hand position, and small dark sphere is the target point. For visualization convenience, 2 driven masses are spatially shifted only for rendering, even physical simulation is done for driven masses that are moved along the same line (that is, they can virtually penetrate through each other). During the course of experiments, the line constraint is horizontal.

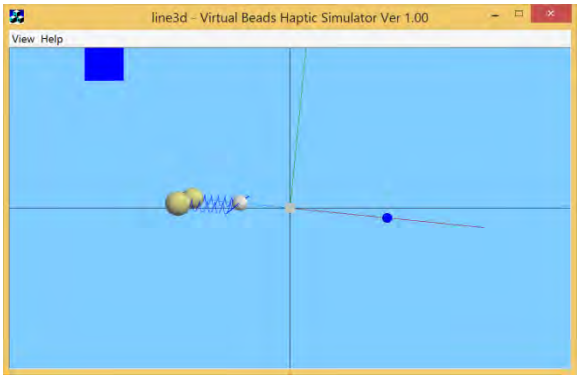


Figure 7. Haptic simulator interface for parallel flexible object.

The square near the top left corner of the GUI window is a semaphore. It provides visual feedback for better motor learning. When trial time is approaching to the described above reaching task time T (with the defined tolerances $\pm\Delta T$), color of the semaphore is changed to green, and if the trial time exceeds maximum $(T + \Delta T)$, color becomes red.

Fig.8 schematically illustrates ergonomics of subjects during the experiments.

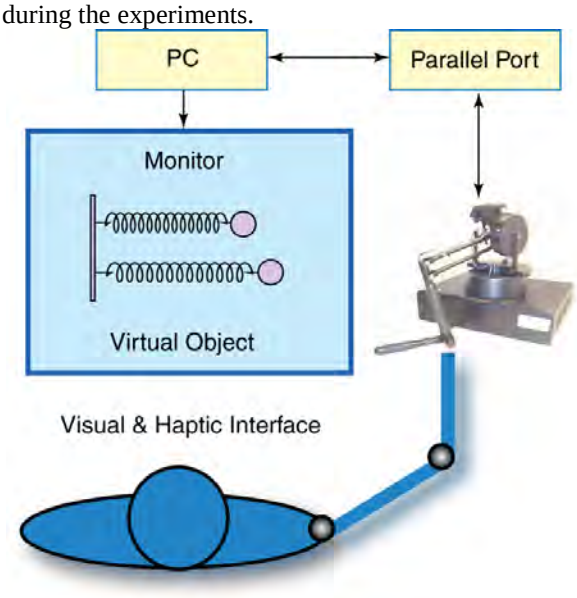


Figure 8. Experimental environment for movement of parallel flexible object.

The experiments were conducted similar to the experimental scheme presented in Section 4, with the following configuration:

$$m_1 = m_2 = 3\text{kg}, \; , k_1 = 50\text{N/m}, k_2 = 250\text{N/m} \; ,$$
$$T = 2.5 \pm 0.3\text{s}, \Delta x = \pm 0.012\text{m}, \Delta v = \pm 0.012\text{m/s}.$$

Fig.9 and **Fig.10** illustrate 5 last trials in experimental series for one of the subjects (thin lines). Thick grey line depicts theoretical velocity profile, and thick black line is the subject's average through all successful trials. Qualitatively, the experimental velocity patterns were similar to theoretically predicted by criterion (1). A quantitative measure for the comparisons was represented by the integrated RMS of the velocity errors,

$$\varepsilon = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(v_{pred}(t_i) - v_{exp}(t_i)\right)^2} \quad ,$$

over the trajectories between the theoretical predictions and the experimental data. Here, N is the number of sampled data in one experimental series (only successful trials are considered). Similar RMS estimator was used for the experiments described in Section 4. Complete analytical solution derivation and experiment description is given in [Svi16].

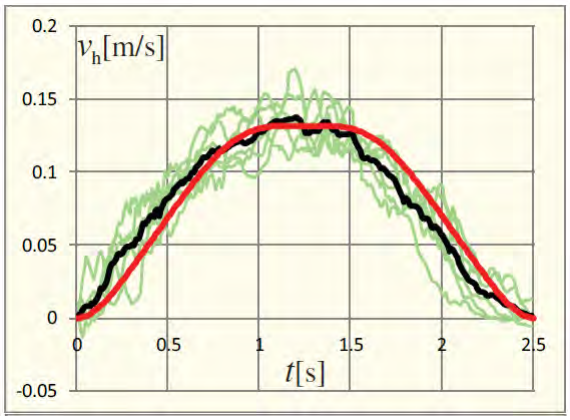


Figure 9. Hand velocity profiles.

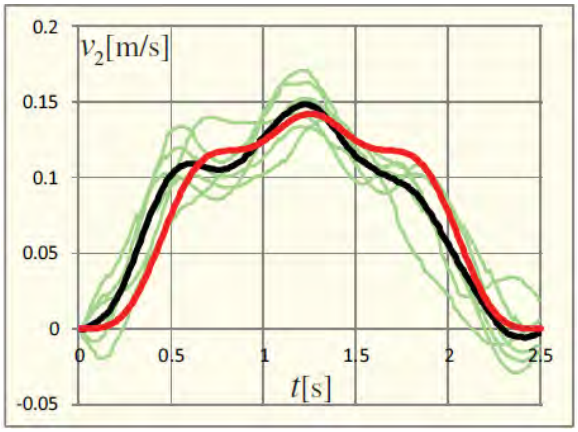


Figure 10. Object velocity profiles (mass 2).

6. CONCLUSIONS

We have discussed a real-time haptic system with interchangeable constraints. The interchangeability of constraints is achieved by a unified interface to link different physical models, basic constraints library processing, and external configuration of the models and associated graphical scenes. This property is required for studying basic constrained human movements, when theoretical movement prediction models should be checked with a large variety of constraints with different shapes, curvatures, viscosity, etc. Several criteria based on optimal trajectory planning were successfully studied with the system for line constraint in 3D for the task of rest-to-rest human movement during transport of flexible object and parallel flexible object.

Experimental data collected with the history unit are clearly in agreement with theoretical results based on the minimum jerk criterion and relating to it variations of the minimum hand force change criterion. This is indirect evidence of the fact that the

human central nervous system plans movements in the task space of hand coordinates. Theoretical velocity profiles correlate well with observed experimental data. Dealing with (parallel) flexible VR objects, subjects after training plan their control strategies to move flexible objects as "a whole", with hand velocity profiles restricted and bell-shaped..

The system facilitates the study of progress in motor movement skills training, when the convergence of hand trajectories to unique and finite profiles observed together with the increase in trial success.

7. REFERENCES

[Bur03] Burdea, G., and Coiffet, P. Virtual reality technology, 2nd Ed. Publ. Wiley-Interscience, 2003.

[Din04] Dingwell, J., Mah, C. F., and Mussa-Ivaldi, F. Experimentally confirmed mathematical model for human control of a non-rigid object. Journal of Neurophysiology, Vol.91, pp. 1158-1170, 2004.

[Fla85] Flash, T., and Hogan, N. The coordination of arm movements: An experimentally confirmed mathematical model. The Journal of Neuroscience, Vol.5, No.7, pp. 1688-1703, 1985.

[Fla03] Flash T., Hogan N., and Richardson M. Optimization principles in motor control. The Handbook of Brain Theory and Neural Networks, 2nd Ed. Arbib M. (ed.). Cambridge, Massachusetts, MIT Press, pp. 827–831, 2003.

[Geo] http://www.geomagic.com

[Gon04] Goncharenko, I., Svinin, M., et al. Cooperative control with haptic visualization in shared virtual environments. Proc. 8th IEEE Int. Conf. on Information Visualisation, London, UK, pp.533-538, July 14-16, 2004.

[Gon10] Goncharenko, I., Svinin, M., Hosoe, S., and Forstmann, S. On the influence of hand dynamics on motion planning of reaching movements in haptic environments. Advances in Haptics, In-Tech Publ., pp.451-462, 2010.

[Kal14] Kaluschke, M., Zimmermann, U., Danzer, M., Zachmann, G., and Weller, R. Massively-parallel proximity queries for point clouds. 11th Workshop on Virtual Reality Interaction and Physical Simulation, VRIPHYS, Bremen, Germany, pp. 19-28, September 24 - 25, 2014.

[Lei12] Leib, R., and Karniel, A. Minimum acceleration with constraints of center of mass: A unified model for arm movements and object manipulation. Journal of Neurophysiology, Vol. 108, No. 6, pp. 1646–1655, September 2012.

[Mor95] Morasso, P., and Sanguineti, V. Self-organizing body schema for motor planning. Journal of Motor Behavior, Vol. 27, No. 1, pp. 52–66, 1995.

[Pir03] Piron, L., Tonin, P., et al. A virtual-reality based motor tele-rehabilitation system. Proc. 2nd Int. Workshop on Virtual Rehabilitation, Rutgers Univ., pp. 21-26, September 21-22, 2003.

[Sal97] Salisbury, J.K., and Srinivasan, M.A. Phantom-based haptic interaction with virtual objects. IEEE Comput. Graph. Appl., Vol. 17, No. 5, pp. 6–10, 1997.

[Svi04a] Svinin, M., Odashima, T., Luo, Z., and Hosoe, S. On the optimization approaches to the trajectory formation of human movements. Proc. Int. Conf. on Complex Systems, Intelligence, and Modern Technology Applications, Cherbourg, France, pp. 628-633, September 19-22, 2004.

[Svi04b] Svinin, M., Masui, Y., Luo, Z., and Hosoe, S. On the dynamic version of the minimum hand jerk criterion. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS2004, Sendai, Japan, Vol. 1, pp. 174-179, September 28-October 2, 2004.

[Svi06] Svinin, M., Goncharenko, I., Luo, Z., and Hosoe, S. Reaching movements in dynamic environments: How do we move flexible objects? IEEE Transactions on Robotics, Vol. 22, No. 4, pp. 724–739, 2006.

[Svi16] Svinin, M., Goncharenko, I., Lee, H., and Yamamoto, M. Modeling of human-like reaching movements in the manipulation of parallel flexible objects. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016), submitted , 2016.

[Uno89] Uno, Y., Kawato, M., and Suzuki, R. Formation and control of optimal trajectory in human multi-joint arm movement: Minimum torque-change model. Biological Cybernetics, Vol. 61, pp. 89-101, 1989.

[Vei09] Veit, A. C. M., and Bechmann, D. Influence of degrees of freedom's manipulation on performances during orientation tasks in virtual reality environments. Proc. 16th ACM Symposium on Virtual Reality and Software Technology (VRST), pp. 51-58, Kyoto, Japan, November 18-20, 2009.

[Wel11] Weller, R., and Zachmann, G. 3-DOF vs. 6-DOF - Playful evaluation of complex haptic interactions. Proc. IEEE Int. Conf. on Consumer Electronics (ICCE), pp. 273-274, Las Vegas, NV, January 9-12, 2011.

[Wol03] Wolfram, S. The Mathematica Book, 5th Ed. Wolfram Media, 2003.

Public Participation to Support Wind Energy Development: The Role of 3D GIS and Virtual Reality

Chen Wang David Miller Iain Brown Yang Jiang
The James Hutton Institute Robert Gordon University
Aberdeen,UK. AB15 8QH Aberdeen, UK. AB10 7QB
{chen.wang, david.miller, iain.brown}@hutton.ac.uk y.jiang2@rgu.ac.uk

ABSTRACT

Wind energy is identified as having a significant contribution to reducing greenhouse gas emission, and Scottish Government targets for the generation of energy from renewable sources. Public policy emphasises the importance of using an ecosystem approach, and the role of public engagement in decisions about future uses of land and sea. A prototype 3D model was developed to present a loch with hypothetical wind turbines on the west coast of Scotland. The model was used to identify issues arising between the growing interest marine renewables, land use changes in line with changing policy and the potential effects on existing seascapes and marine industries and activities. An interface has been developed to provide interactive movement of features in models, including hotkeys to: (i) Switching between images (e.g. 1:50,000 map and aerial images) and GIS Data layers (e.g. Scottish Natural Heritage (SNH) designations); (ii) Introducing new features (e.g. houses, wind turbines, trees); (iii) 'drag and drop' features, guided by the audience.

The virtual reality model was tested with a range of different audience types at events in Oban, on the west coast of Scotland, and Aberdeen on the east coast through Virtual Landscape Theatre (VLT) and Oculus Rift. Factors identified for detailed testing included the significance of lighting conditions on the east and west coast, sea state on perceptions of seascape and wind energy generation, and people's activities at different times of the day.

Keywords

3D GIS, Public engagement, Virtual reality, Wind energy, Virtual Landscape Theatre.

1. INTRODUCTION

In response to development proposals, such as for wind turbines, computer visualisations form part of the materials used in assessing visual impacts. Guidance on standards for such materials, such as the visual representation of wind farms [1], set out requirements and settings for the preparation of visualisations. However, these are restricted to the use of photomontages, and wireframe representations for use in the illustration of the location and nature of a proposed wind farm and predicted visual effects of developments. 3D GIS, VR and associated tools, have benefits which could support delivery of the types of aspirations or regulatory requirements in public policies which relate to planning and development.

ICT tools and visualisation in particular, have been used increasingly as part of information, consultation, and collaboration in relation to issues of global significance. For example, the representation of landscapes of the future including 3D imagery [18][19], sketches, or imagery [20] enables the interpretation of change in relation to landscapes. Visualisation tools have been used for helping communities to plan for adaptation against impacts and effects of climate change as demonstrated by the research team at the Collaborative for Advanced Landscape Planning, at University British Columbia, Canada [21]. They have developed the use of virtual and augmented reality and Geographic Information Systems, with tools such as Community Viz [22], and provide a video game which they describe as empowering lifelong learners to creatively construct their own futures.

Public participation is another issue regarding people engagement in decision making and stakeholder's planning and feedback. The impact on the planning process depends on the level of stakeholder involvement. This involvement can be divided into three aspects as shown by Miller et al [2]:

1) Dissemination, where information is almost exclusively communicated to the public by the 'experts';

2) Consultation, where public opinions are sought and considered in expert or managerial decision-making;

3) Collaboration, where representatives of the public are involved actively in developing solutions and

directly influencing decisions to a greater or lesser degree.

Recently, there is a trend to create methods and tools for investigating landscape and seascape time-depth and historical scenarios through the use of 3D modelling tools and virtual reality engines [11][12], further encouraged by new technological developments that enhance performance and interactivity. For example, virtual reality head-mounted display such as Oculus Rift [14] and PlayStation VR [13] provide a 90 degrees horizontal and 110 degrees vertical stereoscopic 3D perspective. The result is the sensation that you are looking around a very realistic 3D world.

2. BACKGROUND

The 2020 Renewable Routemap for Scotland – Update [3] sets out ambitious targets of the equivalent of 100% of Scottish demand for electricity and 11% of heat capacity to be generated from renewable sources by the end of 2020. This is to be achieved in the context of international agreements for reductions in greenhouse gas emissions alongside those relating to environmental, economic and social considerations. Planning Scotland's Seas [4] notes the importance of considering the onshore implications of offshore developments, and recognises that renewable energy developments offshore have associated infrastructure onshore. In particular, Planning Scotland's Seas identifies links between the marine and terrestrial planning systems, and the requirement for inputs from local stakeholders and knowledge in the development of spatially more detailed Marine Region Plans. It proposes the use of the Ecosystem Approach to better integrate management of seas and coastal areas, the same approach as advocated in the Scottish Land Use Strategy [5].

Currently, 3D based approach plays an important role and makes a real contribution to support wind energy development [23][24]. However, the system developed have lacked direct connection to spatial data, it is a major task to integrate with GIS seamlessly.

In this study visualisation tools were used to present topographic contexts of land and sea use and the introduction of potentially new features and their planning developments such as renewable energy. This takes advantage of the ongoing development of software tools for use in representing 3D environments, such as Maya, 3D Max, Vega Prime, Octaga or specialized landscape visualisation tools such as Visual Nature Studio. These provide a high degree of visual realism for landscape and seascape, enabling the rendering of images or animations [6][7][8][9][10].

3. METHODOLOGY

The main steps involved can be summarised as follows (Figure 1):

(i) Compilation of spatial datasets comprising land sea floor, and surrounding terrain to represent the present-day sea loch;

(ii) Creation of 3D models using existing GIS data, with representation of alternative layouts and designs of offshore wind turbines; 3D models interaction and usability of the interface;

(iii) Development of wind farm preferences using visualisations of each scenario from different viewpoints;

(iv) Elicitation of public opinions on future wind farm planning using VR facilities including VLT and Oculus rift.

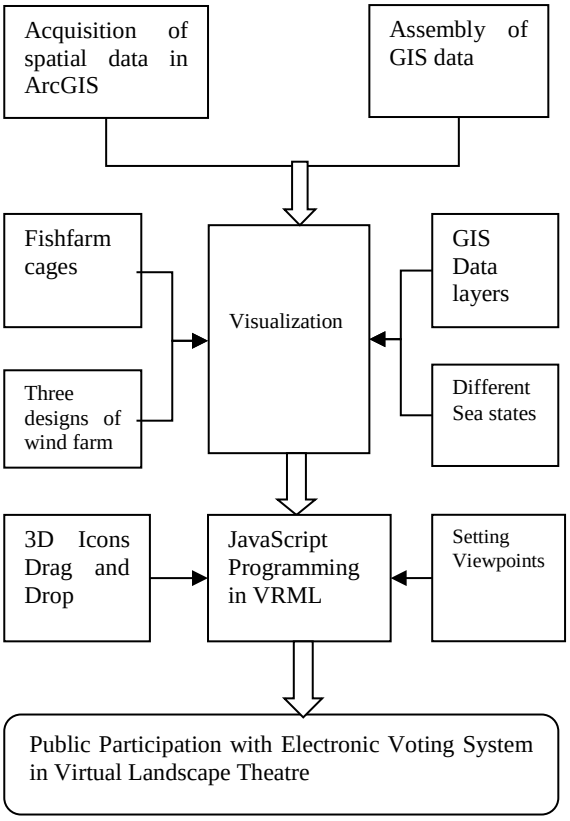


Figure 1 Framework for 3D visualization and simulation of offshore wind farm

The prototype model was used in events designed to elicit public aspirations and concerns regarding future land and sea uses, and to develop scenarios driven by local input. Sessions comprised:

- Introducing drivers of land and sea use change (e.g. movement of features) and electronic voting system;
- Audiences recording preferences for wind farm layout from different viewpoints;
- Audiences voting to prioritise wind farm topics

for in-depth discussions;
- Discussion and voting on sea loch issues (e.g. fish farm location/ size; woodland location/type, building location/type).

3.1 STUDY AREA

The study area was Loch Linnhe, a sea loch on Scotland's west coast, approximately 50 km long, running from Fort William to Oban, at the south end of the Great Glen fault. Land use is dominated by agriculture, particularly crofting on the islands and western shores, forestry, and tourism. Uses of the loch include inshore fishing and fish farming, sailing, and diving, with increasing interest in marine renewable energy.

The topographic context of Loch Linnhe is of glaciated valleys, with terrain rising steeply away from the loch, bare rock and scree, and land on the eastern shore which includes a raised beach. During past ice ages, the loch was a major outlet for glaciers from the Rannoch Moor area, where ice built up in the initial stages of development [15].

3.2 3D Model Creation

A basic model was created of the sea floor and surrounding land of Loch Linnhe in Figure 2.

Image of terrestrial model

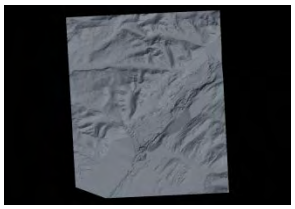


Image of sea bed data

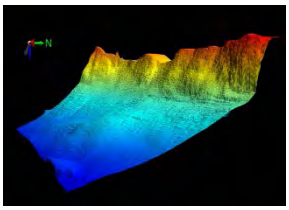


Image of fishfarm cages models

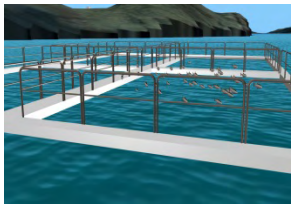


Image of combined model overview

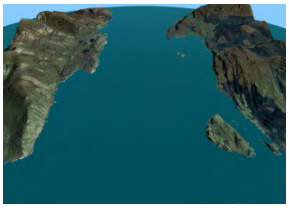


Figure 2 Basic Loch Linnhe Model

This used data of above and below the water line, and the addition of 3D model features, for use in a virtual reality environment:

1) Ordnance Survey (10m resolution) Digital Terrain Model extracted for the land around Loch Linnhe.
2) Multibeam sonar data (1m resolution), surveyed by the UK Marine Environmental Mapping programme (MAREMAP) of British Geological Survey, Scottish Association of Marine Sciences (SAMS) and National Oceanography Centre (NOC), combined with Admiralty seabed data.
3) Autodesk Infraworks used to render a 3D model combining the seafloor and terrestrial areas (221km2; 2.5m resolution), with true scale above sea level and a 2 times vertical exaggeration below sea level.
4) High-resolution aerial imagery used for background landscape textures.
5) Extruded buildings were derived from Ordnance Survey MasterMap.

Further elements added to the model were:

(i) Features associated with coastal environments, developed in Autodesk Maya, including fishfarm cages, leisure craft and renewable energy structures.
(ii) GIS Data layers representing designations (e.g. National Scenic Areas, shell fishing zones).
(iii) Water, using colour to distinguish between above and below water surface.
(iv) Different sea states [16].

A 3D geo-referenced model was created of the island and surrounding sea, with representation of alternative layouts and designs of offshore wind turbines. The spatial data were compiled in ArcGIS, in a single coordinate reference system. The software tools used for the 3D models were Google Sketch-up, Infraworks and Maya.

The spatial data were converted for use in the Octaga virtual reality software in the Virtual Landscape Theatre [11]. The theatre is a mobile curved screen projection facility in which people can be 'immersed' in computer models of their environment to explore landscapes and seascapes.

3.3 Three designs of a wind farm

The model of the windfarm comprises 20 wind turbines, approximately 1.5 km apart, with three different heights of wind turbine: 128m, 165m and 215m. Each turbine is set up with a different location and rotate speed (cut-in wind speed: 3-5 m/s; cut-out wind speed: 25 m/s) [25] in the Loch Linnhe virtual environment which shows a potential area for renewable energy development.

The model includes three different representations of sea state (based on the World Meteorological Organization sea state code; [16]), each with a unique texture and tide height. Wind speed and wind direction have also been considered and corresponding parameters such as cloud cover and wind turbine start-up speed have been added into the

3D Loch Linnhe model, with a dominant wave direction applied to the sea surface, each of which can be switched between. The modelling of illumination conditions is used to enable the inclusion of shadows from the wind turbines in appropriate sunlit conditions and reflections off the sea surface.

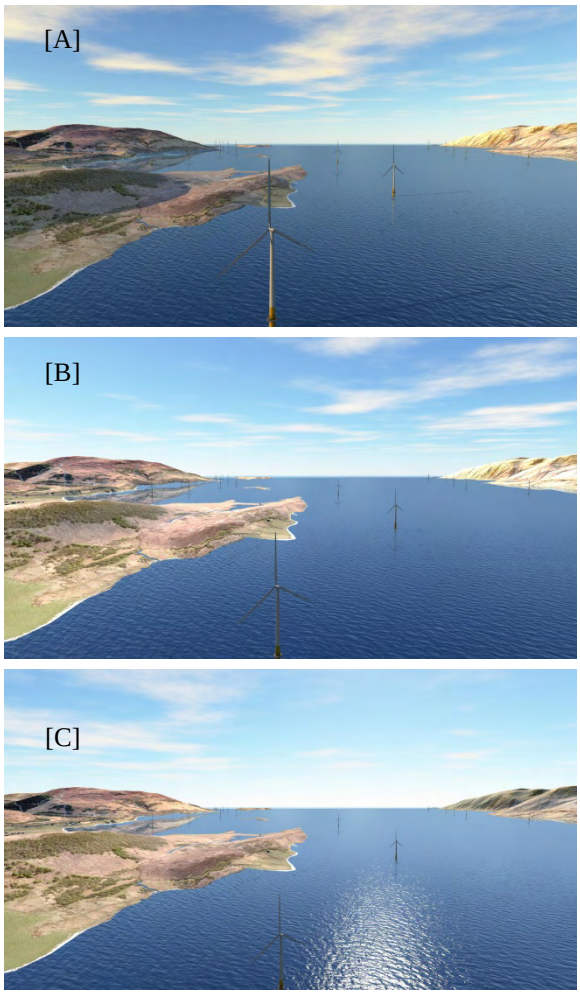


Figure 3 Hypothetical wind farm layouts in 3D virtual Environments of Loch Linnhe: [A] wind turbines 215m to the tip of the blade at sunrise; [B] wind turbines 165m to the tip of the blade at midday; [C] wind turbines 128m to the tip of the blade in the afternoon.

3.4 Interactive functionality and interface for Loch Linnhe model

An interface has been developed to fit with the output of 3D model and the model content with respect to purpose of use. This part of the experiment focused on the interaction and usability of the interface, and the recognizability of the type of visualization. A 'drag-and-drop' feature that allows participants to choose where they would like to position elements (wind turbines, trees, houses, etc.) was added based upon a series of 3D icons (Figure 4). The icons are coded in JavaScript to allow participants to select locations of the forest, housing development, access to the town, car parking, renewable energy, playgrounds and conservation area. It also provides functions for pointing out those areas where audiences definitely do not want such a feature. Icons were 'dragged and dropped' to audience selected positions, with VRML code 'ground clamping' them to the terrain surface (i.e. the icons were automatically located at a vertical elevation consistent with the ground surface).

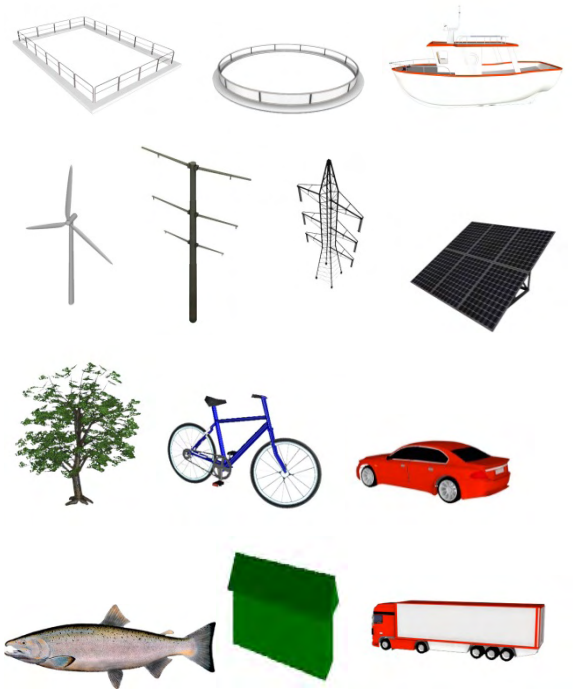


Figure 4 3D icons of land and sea use features

Landscape and seascape could be future modified according to participant's preferences. For example, wind turbines are normally located upon the hill, trees are usually distributed with reference to existing woodland areas, and buildings are mostly situated adjacent to existing settlements. In Figure 5 the infrastructure of an onshore wind farm can be seen. In addition to the wind turbines, the associated power lines for connection to the electricity grid are visible together with features in the vicinity of the development, such as field boundaries and trees. For effective stakeholder engagement it is important to provide sufficient detail of features to enable participants to be able to relate to the site, and locate themselves with respect to a planned development. The level of detail (e.g. number of features, and the visual detail with which they are presented) is tested in a workshop with key stakeholders, so informing the design and implementation of the 3D model.

Figure 6 shows a still from a model of the aquaculture feature, in which the animation version shows the water under different conditions of illumination, levels of variability in waves and the movement of salmon in the cages. Viewing can be from above or below the waterline. When tested with stakeholders (e.g. public, fisherman) at the World Marine Biodiversity Congress, Aberdeen, September 2012, the interactive capabilities were very well received, as was the functionality and appropriateness of the levels of detail.

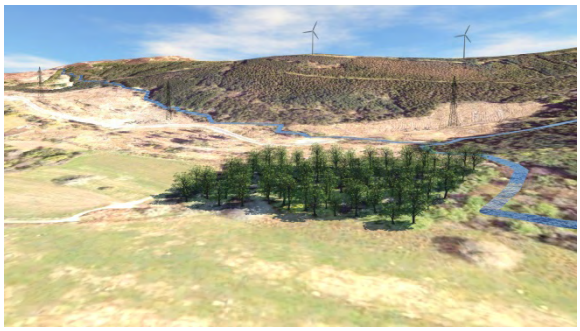


Figure 5 3D drag and drop of icons showing proposed location of new features adjacent to Loch Linnhe.

Figure 6 Planning the siting and support of aquaculture in Loch Linnhe

3.5 Eliciting audience opinions

A prototype of the virtual reality model was tested with a range of different audience types at events in Oban, on the west coast of Scotland, and Aberdeen on the east coast. The Virtual Landscape Theatre and Oculus Rift were used as the medium with invited groups or individual drawn from schools and youth groups, universities, natural heritage managers, planners, and the general public.

The James Hutton Institute's Virtual Landscape Theatre (VLT) is a mobile curved screen projection facility measuring 5.5 metres x 2.25 metres. The screen curvature of 160 degrees provides immersive viewing for up to 20 people. The VLT 'frame' is similar to that used in music concerts, consisting of aluminium trusses which are bolted together to form the walls and roof of the facility. A projection screen is attached to the rear curved wall to form the projection surface. Parallel processing of the 3D models is undertaken by a cluster of three high-end PCs, each consisting of dual quad core processors, RAID Hard Drives and Nvidia FX4800 Quadro graphics cards. The images from each PC is registered and seamlessly joined by 3D Perception UTM before being transmitted to three 3D Perception SX+ projectors. The projectors are mounted overhead and in-front of the screen (front projection). Software models are prepared in either in VRML or OpenFlight formats, and displayed by Octaga Panorama or Vega Prime applications respectively. The portability of the VLT allows it to be used in community venues across Scotland, thereby bringing planning and public participation to planners and the general public. The VLT facilitates visualisation of landscape changes such as woodlands, vegetation, farm management practices, wind farm developments, design and layout of parks, urban expansion and climate change; as well as marine planning such as offshore renewables and aquaculture.

The Oculus rift is one of the low cost HMDs which allow a user to immerse into the virtual environment and look in any direction. It provides an effective resolution of 960 x 1080 pixels per eye with 100° nominal field of view.

Figure 7 Virtual Landscape Theatre: stakeholder dialogue and opinion sharing

Figure 8 Oculus rift: single user exploration of seascapes

Figure 7 and 8 show the use of visualisation tools for eliciting stakeholder opinions, explore scenarios and discussing options. Mobile virtual reality tools for groups (VLT) or individuals (Oculus Rift) are used for selected case study areas.

Table 1 summarises an example of the factors used to assess 'success', and the issues identified by stakeholders as key elements to assess the state of progress of 3D visualisation tools for the purposes of their business needs.

Factors for success	**Issues identified by participant**
Impact	1.Improve efficiency 2.Higher quality decisions 3.Improved communications
Information delivery	1.Multiple modes of information (use of mixed media) 2.Multiple methods of delivery (abstraction, perspectives, interactivity) 3.Multiple interface capabilities (delivery platforms)
Information quality	1.Accuracy 2.Completeness 3.Reliability 4.Unbiased 5.Understandable 6.Relevance
Functionality	1.Trend analysis (prediction) 2.Access to data
Ease of use	1.Easy to use 2.Fast response time (rendering speed)

Table 1 Summary of factors for assessing suitability of VR tools for use by stakeholders.

A regional model was used to introduce the events, providing a context for discussion of issues around the development of offshore windfarms. The Loch Linnhe model was then used to elicit opinions on the issues associated with developing a windfarm in this area of the west coast of Scotland. A preset flythrough route was used to introduce the island, its geography, and the uses of the land and surrounding seas. This was followed by views of the different options for wind farms from specific view points, including at the coast, from specific properties. Audience opinions regarding the views were recorded using electronic handsets.

4. Results

The 3D model and simulation of visual impacts of hypothetical wind farm were used both at events on the west and east coast of Scotland. In total, six formal sessions (108 participants) were arranged for invited groups that consisted of land managers, natural heritage managers, planners, schools and youth groups, university students and the general public.

The medium-scale wind farm was identified as having the strongest preference amongst audience groups (Table 2).

Hypothetical wind farm	Small-scale wind farm	Medium-scale wind farm	Large-scale wind farm
Voting Results (108 participants)	30(27.8%)	58(53.7%)	20(18.5%)

Table 2 Participant preference ratings for hypothetical wind farm in 3D virtual environment.

Audience feedback suggested that the virtual environment was very effective in providing an impression of the different layouts and characteristics of the offshore wind farm, and enabled comparisons to be made of the differences in the visual impacts of the alternative heights of wind turbines (Figure 9).

Figure 9 Audience discussion over the different options for windfarm offshore of Loch Linnhe

Comparing the feedback on presentations in venues on the west and east coast of Scotland, with models of windfarms on each coast, the issues arising included the different impacts in the morning and evenings of developments on the east and west coast relating to lighting conditions and the patterns of people's daily activities. In particular, differences were identified between visual impacts at sunrise and sunset in an east and west coast environment, and the effects of horizontal views (i.e. with sky backdrops) compared to those downwards towards the development (i.e. with sea backdrops).

Findings from use of the prototype are being used to develop tests to consider the potential significance of sea state with respect to view characteristics, and the significance of different lighting conditions and turbine layouts on people's landscape and seascape preferences [17].

5. Discussion and Conclusions

The nature of audience interaction with the models appears to have been appropriate to satisfy the aims of the participation. Based on voting results from feature recognition (e.g. lochs, islands, mountain, villages, woodlands) and hypothetical wind farm layouts, the virtual environment provided materials with levels of familiarity suitable for credible suggestions for consideration of existing and new features. Audience surveys suggest that the package (i.e. the evidence of recording views, relevant models, the facility and its interactivity) supported material participation, beyond that of information dissemination. The level of influence on final decisions remains to be assessed after completion of the process of plan development.

Exploring and interpreting the offshore environment was reported by teachers, and professionals, as providing a better understanding of the potential impacts of a proposed windfarm. Some of the issues raised were identified as being of specific relevance to the school curriculum for follow-up discussions in class. Feedback from professionals in natural heritage management and planning reported the value of being able to see representations of different options in heights and siting of turbines, and from locations selected by members of the audience.

Engaging with stakeholders and the public has enabled discussion, explanations and opinions to be exchanged, and feedback on renewable energy use, now and in the future. The results are being used to inform improvements in the design of tools for eliciting public responses to prospective changes in offshore wind farms interpretation, and demonstration of one aspect of an ecosystem approach to the planning of change at sea.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the funding for this work from the Scottish Government Rural Environment Scientific and Analytical Services. Ordnance Survey data were obtained through the One Scotland Mapping Agreement with Scottish Government.

7. REFERENCES

[1] SNH 2014. Visual representation of wind farms. Scottish Natural Heritage. http://www.snh.gov.uk/publications-data-and-research/publications/seitarch-the catalogue/publication-detail/?id=846

[2] Miller, D. R., Morrice, J., Horne, P. and Ball, J, "Integrating programmes of awareness and education for professionals and the Public", proceedings of Environment 2007, Abu Dhabi, 27 January – 2 February, 2007.

[3] Scottish Government 2012,2020 Renewable Routemap for Scotland – Update www.scotland.gov.uk/Resource/0040/00406958.pdf

[4] Scottish Government (consultation documents) 2013, Planning Scotland's Sea: Scotland's National Marine Plan (including offshore renewables) www.scotland.gov.uk/Topics/marine/marine-consultation

[5] Scottish Government 2011, Land Use Strategy, Scottish Government. www.scotland.gov.uk/Resource/Doc/345946/0115155.pdf

[6] Ball, J., Capanni, N. and Watt, S. 2007, Virtual reality for mutual understanding in landscape planning. International Journal of Social Sciences 27(2) pp78-88.

[7] Wang C., Wan T.R and Palmer I. 2010, Urban Flood Risk Analysis for Determining Optimal Flood Protection Levels Based on Digital Terrain Model and Flood Spreading Model, The Visual Computer, Springer, 26 (11): 1369-1381.

[8] Wang. C, Miller. D, Jiang Y and Donaldson-Selby. G, Use of 3D Visualisation Tools for Representing Urban Greenspace Spatial Planning, 2015 IEEE International Conference on Information Science and Control Engineering (ICISCE 2015), 24-26 April, pp 528-532, 2015.

[9] Resch. B, Wohlfahrt. R and Wosniok. C, Web-Based 4D Visualization of Marine Geo-Data Using WebGL, International Journal of Cartography and Geographic Information Science, 41:3, 235-247, 2015.

[10] Wang. C, Wan, T.R. and Palmer, I.J. 2012. Automatic reconstruction of 3D environment using real terrain data and satellite images. Intelligent Automation and Soft Computing, TSI, 18(1), 49-63.

[11] Vasáros, Z. 2008. Authenticity and accuracy of virtual reconstructions – a critical approach. In: CAA2008 Session – On the Road to Reconstructing the Past, Programs and Abstracts, Budapest, Hungary, April 2–6, pp. 249. ISBN: 978-963-8046-95-6.

[12] Verhagen, P. 2008. Dealing with uncertainty in archaeology. In: CAA2008 Session – On the Road to Reconstructing the Past, Programs and Abstracts, Budapest, Hungary, April 2–6, pp. 99. ISBN: 978-963-8046-95-6.

[13] https://en.wikipedia.org/wiki/PlayStation_VR

[14] https://en.wikipedia.org/wiki/Oculus_Rift

[15] http://planetearth.nerc.ac.uk/features/story.aspx?id=749

[16] WMO 2008, Guide to Meteorological Instruments and Methods of Observation) - part II, Chapter 4 (Marine Observations), publication No. 8.

[17] Bishop, I.D. and Miller, D.R. 2007, Visual assessment of off-shore wind turbines: the influence of distance, contrast, movement and social variables *Renewable Energy* **32** pp 814–831.

[18] DOCKERTY, T., LOVETT, A., APPLETON, K., BONE, A., SUNNENBERG, G. 2006. Developing scenarios and visualisations to illustrate potential policy and climatic influences on future agricultural landscapes. Agriculture, Ecosystems and Environment, 114, 18.

[19] Donaldson-Selby, G.; Wang, C.; Miller, D.R.; Horne, P.; Castellazzi, M.; Brown, I.; Morrice, J.; Ode-Sang, A., Testing public preferences for future land uses and landscapes (2012) GIS Research UK Conference 2012, University of Lancaster, April 2012.

[20] Palomo, I., Martín-López, Berta., López-Santiago, Cesar., Montes, Carlos.,2011. Participatory scenario planning for protected areas management under the ecosystem services framework: the Don˜ana socialecological system in Southwestern Spain. Ecology and Society, 16.

[21] SHEPPARD, S. R. J., SHAW, A., FLANDERS, D., BURCH, S. & SCHROTH, O. 2013. Bringing climate change science to the landscape level: Canadian experiences in using landscape visualisation within participatory processes for community planning. . In: FU, B. & JONES, K. B. (eds.) Landscape ecology for sustainable environment and culture. Dordrecht: Springer Netherlands.

[22] PLACEWAYS. 2013. Community Viz [Online]. Available: http://placeways.com/communityviz/.

[23] Bishop, I. and Miller, D.R. 2007. Visual influence of off-shore wind turbines: the influence of distance, contrast, movement and social variables. Renewable Energy 32**,** 814-831.

[24] Manyoky, M., Wissen Hayek, U., Heutschi, K., Pieren, R. and Grêt Regamey, A. (2014) Developing a GIS-Based Visual-Acoustic 3D Simulation for Wind Farm Assessment. ISPRS International Journal of Geo-Information, 3, 29-48. http://dx.doi.org/10.3390/ijgi3010029

[25] Siemens Wind Power Platform , http://www.energy.siemens.com/hq/en/renewable-energy/wind-power/

Framework for Automated Customer Service in Sign Language

Filip Malawski
AGH University of Science and Technology
Department of Computer Science
Kraków, Poland
fmal@agh.edu.pl

Jakub Gałka
AGH University of Science and Technology
Department of Electronics
Kraków, Poland
jgalka@agh.edu.pl

ABSTRACT

Deaf people need the help of an interpreter in formal relations, such as visiting offices or medical institutions. We present a new framework for building systems for sign language interaction, which can provide basic automated customer service for the deaf. The framework covers all steps required to build such a system from scratch - the acquisition of scenario-specific corpora, extraction of features, training of models, recognition, user interface, integration and configuration of the final application. The usability of the framework has been evaluated by creating a proof-of-concept system for automated scheduling of doctor's appointments in sign language. The results indicate that the process of building a sign language interaction system with our framework is relatively quick and simple. Recognition efficiency was evaluated as well and proved to be sufficient for practical use.

Keywords

Sign Language Recognition, Kinect 2, HMM, Parallel HMM, HCI, Framework

1. INTRODUCTION

It is difficult for the deaf to function normally in society, as they communicate mainly by using sign language (SL), which most people do not know. Writing is not a viable solution, for several reasons. First of all, due to their different education levels, the deaf often do not know how to read and write. The grammar of a spoken and written language is considerably different from the grammar of sign language, therefore it is not easy for them to learn it. Moreover, even those who can read and write consider using sign language to be much faster and much more natural.

This problem with communication has an impact not only on the informal relations between deaf people and healthy persons, but in formal cases as well. In particular, when going to places such as offices or medical institutions, the deaf must rely on an interpreter. Some institutions provide translation services, and in other cases they need their own interpreter. The translation services offered by institutions can take the form of a special appointment with an interpreter on site or a web-conference with a remote interpreter. In either case, providing proper customer service for the deaf is time-consuming and expensive.

In some cases there is yet another concern – deaf people may want to keep their issues private, without involving another person as an interpreter. Particularly in the case of medical institutions, even making an appointment with certain types of doctors may be uncomfortable for deaf patients.

Automatic recognition and translation of sign language could significantly help in addressing the aforementioned issues. It would be less expensive than live interpreters, as well as easily and constantly available and comfortable for the deaf. The main concept of our framework is to provide various institutions with the means of creating a dedicated system, specific for their scenario, which would provide the deaf with at least basic customer self-service without any help from a live interpreter. The idea is similar to Interactive Voice Response (IVR) systems used in call centers. The customers are serviced based on the recognition of user sign-language gestures and pre-recorded or synthesized messages (prompts). Typical cases may be handled automatically and in non-typical ones it is possible to switch to a live interpreter.

We present a framework which allows for a quick and easy creation of automated customer service systems in sign language. It covers all necessary elements, from recording scenario-specific corpora, through data processing, training of models for the recognition module, front-end user application and the configuration of the whole system.

Data acquisition and processing is based on depth-image (Kinect 2) sensor, which provides the recognition of the user pose in the form of a fitted skeleton model. Sign language recognition is based on a Parallel Hidden Markov Model (PaHMM) classifier. The tools for the acquisition of structured corpora, training of models and the configuration of the final system are provided, together with a simple user application. The framework can be easily modified by the substitution of any of the modules. Therefore it is flexible and adaptable not only to various usage scenarios, but to different data sources, processing and recognition methods as well.

The evaluation of our framework was performed by creating a system which allows for automatic scheduling of doctor's appointments using sign language. We chose this particular area due to the results of a questionnaire among deaf people, in which they were asked to indicate situations where such a system would be most useful for them. We cover a simple scenario, consisting of a two step dialog – the selection of the doctor and the day of the appointment. Using our framework we recorded a database of 14 gestures for different doctors and 10 gestures for selecting days in Polish Sign Language. With these data, we trained the models for recognition, configured the dialog flow-chart and finally obtained a working prototype of the system. Based on this use-case we evaluated the usability of the framework and the efficiency of the recognition process. The initial results and user feedback are encouraging.

2. RELATED WORK

Automatic sign language recognition is recently a much-investigated topic. State-of-the-art works differ mainly in terms of data modalities, processing and recognition methods. On the other hand, practical deployments and usage are rarely addressed as research objectives.

Considering data modalities and processing, multiple approaches can be found in literature. Although some works depend solely on RGB data [ThPM14, YaSL10], the usage of depth sensors, such as the Kinect, proved to be greatly beneficial [DoDZ13, KaKh15]. Depth data allows for easy and efficient extraction of the person in the image, as well as body segmentation and tracking. Moreover, the Kinect provides skeleton data, which by itself is often reliable enough for efficient gesture recognition [ASSM16, ISTC14]. In our work, we employ Kinect 2 skeleton data, using normalized positions and orientations of selected joints.

Various machine-learning methods have been employed for the recognition process. The most popular approaches use Dynamic Time Warping [BaDr13, JaKh14, MQSA14] and Hidden Markov Models (HMM) [GFZC04, LiKK16, VeAC13]. Both were shown to provide high accuracy of recognition [RMPS15]. Other approaches include: Support Vector Machines [KoRa14], Neural Networks [AnKS15] and Random Forests [RAWD13]. We employ Parallel HMM, which we found to be superior to the traditional HMM method especially in terms of higher robustness to feature distortions and better classification confidence levels than in classical HMMs. Similar conclusions are reported in [ThPM14].

One particular issue with sign language recognition is that there are different languages in each country, just as in the case of spoken languages. Usually each research team works with SL native to their country – there are works on American SL[DoLY15], Arabic SL[ASSM16], Chinese SL [WCZC15], Indonesian SL[RAWD13], etc. This makes the comparison of results difficult, as there is no international SL database. The situation is no different in our case – we conduct experiments with a custom database, recorded by us specifically for our scenario, using

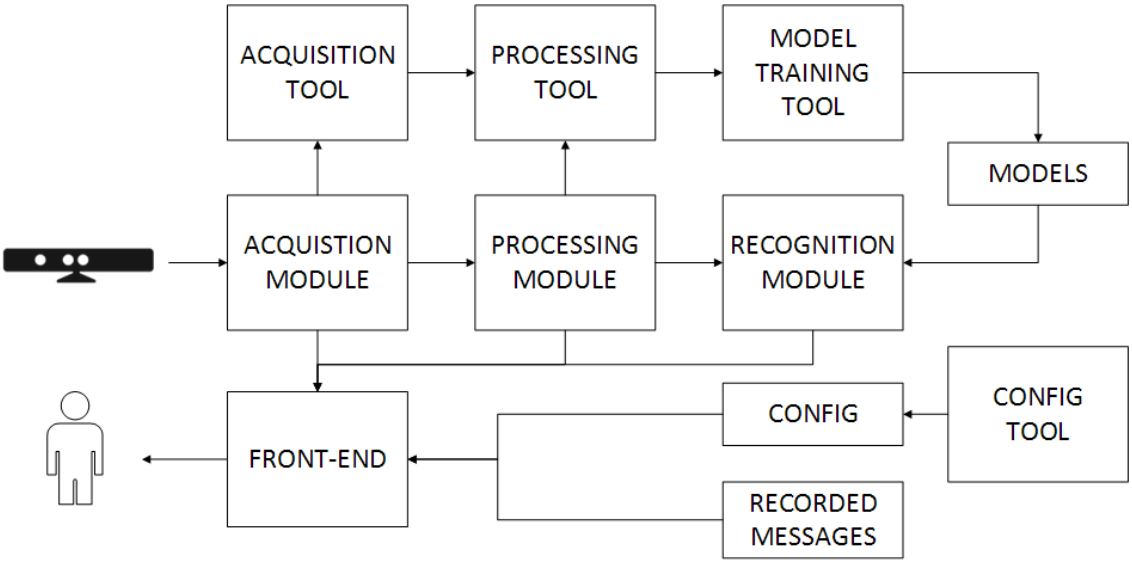


Figure 1: Framework architecture

Polish SL. So far Polish SL has been analyzed in [OsWy13]. It is also important to note, that we report results obtained using a practical system.

As mentioned before, most papers focus on data processing or machine-learning methods, while little work has been done considering the practical usage of these methods. Contrary to that, our framework's main focus is on making these methods available for real-life responsive applications.

3. PROPOSED FRAMEWORK

Overview

Our framework aims at facilitating the creation of interaction systems where the general idea is to display questions to the user and recognize his answers, recorded by a dedicated sensor, such as Kinect 2. Currently, in order to improve the accuracy of the system, the user is actively prompted to answer with one word or phrase only. Therefore, the dialogs must be constructed accordingly, with a set of answers expected by the recognizer grammar related to the specific prompt. Nevertheless, even building such a relatively simple interaction system is challenging and our framework focuses on simplifying the development process.

The architecture of our framework is presented in Figure 1. Core elements include the acquisition, processing and recognition modules. The acquisition module interfaces with the sensor and provides the data. The processing module extracts specific features from raw data. The recognition module classifies a given sample by using previously trained models. These core elements are used by the tools and the front-end application.

The acquisition tool is a dedicated application which allows for a quick and easy recording of a specific database of gestures. The processing tool applies any feature extraction method chosen from the processing module to a given database and produces files with computed features. The model training tool generates models for specific sets of gestures, based on the feature files. The models are then used by the recognition module. The configuration tool provides a user-friendly graphical interface for defining dialogs and editing paths in the configuration file.

The front-end application integrates the acquisition, processing and recognition modules and provides a simple user interface for interaction with the system. It requires recorded messages which are to be displayed. Alternatively these messages can be synthesized by employing a virtual signing avatar, although this feature is still being developed. The settings of the front-end application (paths to models and recordings, dialog flow-chart, etc.) are defined in the configuration file.

Two programming platforms are employed in the current implementation, namely MATLAB and .NET. The front-end application, acquisition, processing and configuration tools as well as all three core modules are implemented in .NET. The model training tool is implemented in MATLAB, as it allows for relatively quick verification of different approaches. Both the recognition module and the model training tool employ an external implementation of HMM, namely HTK[1].

The elements of the framework which are loosely coupled communicate by creating specific files. The acquisition tool produces avi files with RGB and depth data as well as MATLAB .mat files with skeleton data. The processing tool produces .mat files with the features. We employ .mat files for storing both the skeleton data and the features, since this format can be easily handled not only in MATLAB, but in .NET as well, due to a dedicated library. This also facilitates the potential substitution of the elements of the framework. The models are saved in an HTK-specific format and the configuration file uses the XML format. Communication between the tightly-coupled elements (core modules, front-end application) employs dedicated .NET classes.

Acquisition and processing modules

In current implementation of the acquisition module we employ the Microsoft Kinect 2 sensor. It provides multiple data streams, namely RGB images, depth images and skeleton data. The Kinect 2 skeleton model consists of 25 joints, which is an improvement compared to the first version of the Kinect, which provided only 20 joints. Additional joints include the spine between the shoulders, the tip of the left and right hand, and thumbs. The newly added tip and thumb joints are particularly interesting in the context of gesture recognition. For each joint x, y, z the coordinates in the camera frame-of-reference space are provided. Additionally, for all joints except for the tips and thumbs, orientations are given in the form of quaternions.

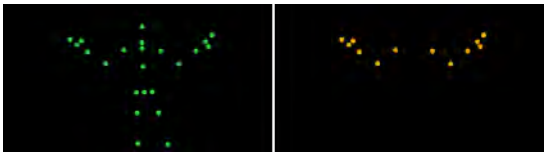


Figure 2. Kinect 2 skeleton joints (sitting person). Left: all tracked joints. Right: joints selected for sign language recognition.

We decided to use skeleton data, as it enables relatively accurate tracking of joint trajectories during signing. Moreover it is robust to illumination and background changes. We identified all joints from both hands and arms to be the most relevant for

[1] http://htk.eng.cam.ac.uk/

gesture classification. The selected joints include (for each hand): shoulder, elbow, wrist, hand, tip, and thumb (see Figure 2). We use x, y, z coordinates for all selected joints as well as all available orientations. In order to better adapt the system to various users, the positions of joints are normalized relative to the position of the head. Many gestures in sign language are performed in a specific position relative to the head, e.g. a dentist requires performing the gesture near the mouth. Therefore the head was chosen as a good reference point for the other joints. After normalization, the first and second derivatives of the joint positions are computed as additional information for the recognition module.

Recognition module

Hidden Markov Models are often used for gesture recognition, due to their high efficiency in temporal pattern recognition. The idea of HMM is to build a model of hidden states, based on known observations. Parallel HMM contains multiple channels, each with its own model. During the classification process, outputs from all channels are combined into a single result. There are multiple approaches to both grouping features into channels and combining the results.

In our system we employ PaHMM with grouping based on the skeleton joints. The positions and orientations of each joint form separate joint-feature channels for the PaHMM (see Figure 3). This method of grouping features corresponds directly to the nature of the modeled phenomenon - various gestures engage various joints to a different extent. Particularly, some gestures differ in the positions of joints during movement, while other ones differ only in orientations.

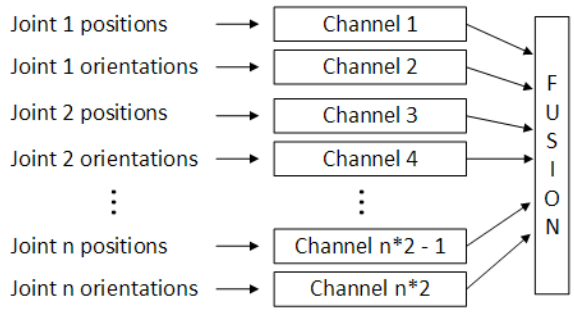


Figure 3: Parallel HMM channels

We perform a weighted fusion of the modeling results from all channels by assigning fusion weights during training. The classification is then performed based on the weighted log-likelihood channel fusion. The weights are computed based on the accuracy of each separate channel. For the sake of comparison we also evaluate the database with a classical, single channel full joint-feature HMM.

The key assumption in our framework is that each question (system prompt) has its own set of expected possible answers and therefore, for each question, we train a separate recognition model. This allows to achieve high recognition efficiency while providing a convenient interaction method for the users.

Tools

Our framework includes a number of tools, which are not directly used in the final front-end application, but are essential in adapting it to a given scenario during deployment phase.

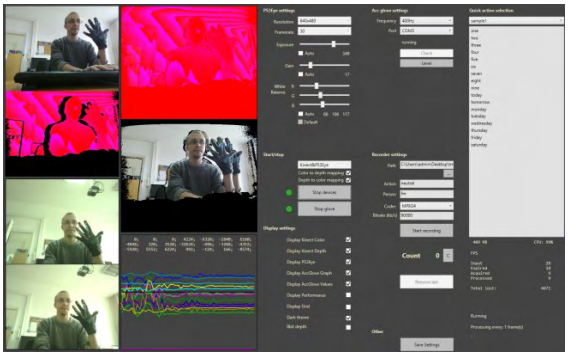


Figure 4: Acquisition tool

A proper amount of data is crucial for the recognition and the database acquisition step can be optimized only by efficient recording procedures. For the purpose of effective data acquisition we have created dedicated software (see Figure 4), which allows to record data from Kinect 2. The operator needs only to press 'start/stop' button, therefore during the recording sessions only a few seconds are needed for each sample. The data is recorded in a format and structure compatible with the rest of the framework and can be directly sent to the processing tool.

Although we currently employ skeleton data only, the software can record all Kinect 2 data streams. Moreover it allows for the acquisition of data from other devices, namely two PS3Eye cameras and an accelerometer glove and provides data stream synchronization mechanisms. These additional data may be used in different approaches or deployment scenarios. The software and its source code are freely available under the GPL license[2].

The processing tool allows to easily process an entire recorded database with a method selected from the processing module and to produce a properly structured database of feature files. The model training tool employs these features to generate Parallel HMM models, which can be directly used in the recognition module. The training process is parallelized in order to accelerate this step in case of multi-core CPUs. The algorithms in this tool can be modified in order to validate different learning and

[2] https://github.com/fmal-pl/MultiSourceAcquisition

recognition approaches. Finally, the configuration tool allows to easily edit the XML configuration file, which defines the dialog, as well as paths to models and recordings, etc. It provides a convenient graphical tool for editing the conversation flow-chart (see Figure 5).

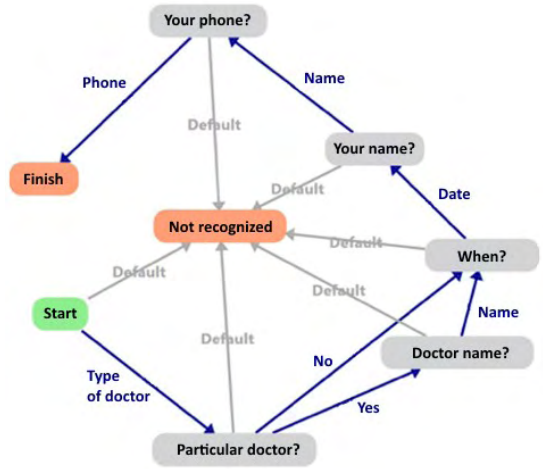


Figure 5: Dialog definition flow-chart interface in the configuration tool

Front-end

The translation site is equipped with a display, a computer and the Kinect 2 sensor mounted below or above the display. The user stands or sits in front of the display and interacts with the system (see Figure 6). The user interface of the front-end application has a single window, where recorded messages and video feed from the camera are displayed interchangeably. The application can also display subtitles, mostly for the sake of demonstrational purposes, for people with poor or no knowledge of sign language.

Figure 6: Translation site

The interaction consists of a series of steps. In each step, a message ending with a question is displayed and the user is prompted to respond by using sign language. During the response stage, the video feed from the Kinect RGB camera is displayed, so that the user can see how his answer is recorded. The time available for answering is set by default to 3 seconds, although in can be configured differently.

The response is recorded, processed and classified. The system has a database of recorded messages with all available answers for each question. It displays a message corresponding to the recognized gesture as feedback to the user and then proceeds to the next question accordingly to the conversation flow-chart. The system may ask the user to repeat the gesture if the confidence of the recognition is below the expected threshold. It prevents false positives and increases system reliability significantly. In the case of multiple failed recognitions the system displays a message asking to switch to a live interpreter and ends the interaction. In case when answers to all questions are recognized an end message is displayed after the last question and the system is ready to send the gathered information.

4. EVALUATION

Proof-of-concept

We evaluated our framework by using it to implement a proof-of-concept system for the automatic scheduling of doctor's appointments using sign language. The scenario assumed a simple, yet functional dialog, where the user is asked which doctor she or he would like to see and when.

For this scenario we recorded a database of 24 different gestures, signed by 7 persons, who were coached and supervised by a professional signer. The database consists of 14 gestures associated with various medical specializations (allergist, cardiologist, dentist, dermatologist, diabetologist, dietician, gastrologist, gynecologist, laryngologist, ophthalmologist, oncologist, psychologist, psychiatrist, and surgeon), and 10 gestures associated with selecting days (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, today, tomorrow, and day after tomorrow). Figure 7 presents the key frames from some of the recorded signs. Each gesture was repeated by each person 10 times. The dataset will be available for academic community since October 2016 (please contact the authors for details).

The two categories of gestures included in the database (doctors and days) were chosen not only because they fit the scenario, but also due to their distinct characteristics. In the case of doctors the gestures are considerably different from each other, often performed using two hands and with various kinds of hands movements (see Figure 7, top row). Gestures for days, on the other hand, are very similar. In some cases they differ only in the positioning of the fingers, while the movement of the hands remains the same (see Figure 7, bottom row). This constitutes a much more difficult classification problem, in particular for the methods based on Kinect 2 skeleton data, which contain the positions of only two fingers (index and thumb) and even so, these data are often

not quite accurate. Therefore different recognition accuracy is to be expected for each category - an issue discussed in the recognition evaluation section.

The recorded database was run through the processing tool in order to extract the features - the normalized positions and orientations of selected joints. In the next step, the model training tool was employed to prepare the models (separate for each category) for the recognition module. The dialog was defined in the graphical editor of the configuration tool and the video messages required for the dialog were recorded using a standard camera. Finally, the remaining part of the configuration was prepared (paths to models and recorded messages) and uploaded to the front-end application.

Figure 7: Key frames of selected gestures from recorded evaluation database. Top row: allergist, dermatologist. Bottom row: Tuesday, Wednesday

Usability

In this section we evaluate the usability of the framework in context of the task of preparing a proof-of-concept system, by analyzing each step of the process.

The time needed for a recording session with one person was approximately one hour, including teaching the person each gesture prior to signing it. Therefore, the acquisition of the entire database of 24 gestures with 7 persons required approximately 7 hours. More complex scenarios would naturally require more time to record all of the necessary gestures. Compared to our previous experiences with general purpose recording software, our dedicated acquisition tool significantly reduced the time needed for both recording and preparing the data for further processing.

In order to extract the features, the processing tool required only to enter the path to the recorded database and the path to the main directory of the feature files. The extraction lasted only a few minutes, since the processing of skeleton data is not very time-consuming. The model training tool operated in a similar fashion – it required input and output paths and also a division of the dataset, since doctors and days were trained as separate models for separate questions in the dialog. We used our default settings for the learning process, although depending on the approach, the configuration of the model training tool may require selecting the proper parameter values. The time needed for training both models was approximately 1 hour on a computer with a 4-core 3.8GHz processor.

Recording the video messages for the dialog (with a standard camera) took a approximately 2 hours, mostly due to multiple repetitions required to achieve satisfactory quality of each video. The configuration of the dialog and the remaining settings of the front-end application took about an hour. The front-end application, although simple, was positively rated by two professional signers, who found the system to be convenient and innovative.

Summarizing, the framework allowed to quickly and easily create a functional system for basic customer-service in sign language. In the case of the proof-of-concept system it took only several hours to record the database and another few to setup the rest of the system. All steps were easy to perform and required little specific knowledge. Although complex scenarios would naturally require more work, we conclude that our framework greatly facilitates the creation of such systems.

Recognition results

Proper recognition efficiency is essential for the practical use of interaction systems build with our framework, which is why it was also analyzed. As mentioned before, we trained separate models for each question, therefore the evaluation of recognition was performed separately for both categories (doctors and days). For comparison, we used both HMM and PaHMM classifiers.

We evaluated only the user-independent case, as this is relevant to our practical application. We used the leave-one-out cross-validation scheme - samples from each person were tested with a model trained on all the other persons. Since we have 7 persons in the database, there were 7 folds of the cross-validation. The presented results were averaged from all folds.

We employed multiple performance measures in order to be able to better assess how our system would suit the given scenario. They are based on the following values: P (positive) – number of examples in the given class, N (negative) – number of examples in other classes, TP (true positive) – number of correctly classified positive examples, TN

- number of correctly classified negative examples, FP - number of negative examples classified incorrectly as positive examples, FN - number of positive examples classified incorrectly as negative examples Employed performance measures include:

$$accuracy = \frac{TP + TN}{P + N} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

We also use error equal rate (EER), which is the rate at which both the false acceptance and false rejection rates are equal.

Results for the 'doctors' category are given in Table 1 and for the 'days' category in Table 2. These include both HMM and PaHMM results.

%	Acc	EER	F1	Prec.	Recall
HMM	83.16	5.40	83.41	83.64	83.17
PaHMM	91.22	4.08	91.22	91.22	91.23

Table 1. Results for 'doctors' category

%	Acc	EER	F1	Prec.	Recall
HMM	67.00	14.95	66.95	66.90	67.00
PaHMM	75.57	9.92	75.51	75.46	75.57

Table 2. Results for 'days' category

Several conclusions may be drawn from the results. First of all, there is a considerable discrepancy between the results for the 'doctors' and 'days' category. As indicated in the database description, 'days' gestures were expected to be a more difficult case due to their similarity. As they differ mostly in the positioning of the fingers, there is a clear need to enhance the recognition process with features corresponding to hand shapes in order to properly handle this type of gestures. On the other hand, Kinect 2 skeleton data is sufficient to achieve satisfactory results for hand-shape-independent gestures.

In both categories, the results for PaHMM are superior to the classical HMM. The absolute difference in accuracy, precision, recall and F1 score is approximately 8%, which constitutes a significant improvement. Similar values of precision, recall and F1 score indicate that the recognition model is well balanced. Low EER is particularly important in case of practical applications. It indicates that the confidence threshold of the classifier may be set in such a way that almost all false positives are rejected, while at the same time almost all true positives are accepted. This corresponds directly to the usability of the final system. In the case of the 'doctors' category the EER is sufficiently low to put the model into a practical use. In the case of the 'days' category, as mentioned before, hand shape features need to be added to achieve comparable results. It is worth mentioning, that the results were obtained with relatively small database, and increasing number of subjects and samples is likely to improve recognition efficiency as well.

5. CONCLUSIONS

We presented a complete framework for building sign language interaction systems for providing basic customer service for the deaf. We evaluated the feasibility and usability of the framework by creating a simple system for making appointments with a doctor in sign language. We concluded that the framework enables easy and quick creation of sign language interaction self-service systems, while providing significant use case flexibility. It can be easily adapted to different scenarios and different recognition approaches. The recognition efficiency indicates that in the case of the 'doctors' gestures, which are not strictly dependent on hand shapes, the results are sufficient to put the model into a practical use. Low EER corresponds to the high usability and robustness of the final system. In the case of the 'days' gestures, additional features are required in order to handle the different hand shapes better. In the future we intend to add hand shape descriptors extracted from depth images and also extend the framework with a virtual signing avatar, which would provide an alternative to the pre-recorded video messages.

6. ACKNOWLEDGMENTS

This work was supported by the Polish National Centre for Research and Development - Applied Research Program under Grant PBS2/B3/21/2013 titled "Virtual sign language translator".

7. REFERENCES

[AnKS15] Anand, V., Keskar, A. G., and Satpute, V. R.: Sign Language Recognition Through Kinect Based Depth Images And Neural Network. 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 194–198, 2015

[ASSM16] Amin, O., Said, H., Samy, A., and Mohammed, H. K.: HMM based automatic Arabic sign language translator using Kinect. 10th International Conference on Computer Engineering and Systems, ICCES 2015, pp. 389–392, 2016

[BaDr13] Barczewska, K. and Drozd, A.:

Comparison of methods for hand gesture recognition based on Dynamic Time Warping algorithm. 2013 Federated Conference on Computer Science and Information Systems, pp. 207–210, 2013

[DoDZ13] Dominio, F., Donadeo, M., and Zanuttigh, P.: Combining multiple depth-based descriptors for hand gesture recognition. Pattern Recognition Letters, Elsevier B.V., 2013

[DoLY15] Dong, C., Leu, M. C., and Yin, Z.: American Sign Language Alphabet Recognition Using Microsoft Kinect. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 44–52, 2015

[GFZC04] Gao, W., Fang, G., Zhao, D., and Chen, Y.: A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognition vol. 37, Nr. 12, pp. 2389–2402, 2004

[ISTC14] Ibañez, R., Soria, Á., Teyseyre, A., and Campo, M.: Easy gesture recognition for Kinect. Advances in Engineering Software vol. 76, pp. 171–180, 2014

[JaKh14] Jambhale, S. S. and Khaparde, A.: Gesture recognition using DTW & piecewise DTW. 2014 International Conference on Electronics and Communication Systems, ICECS 2014, pp. 1–5, 2014

[KaKh15] Kane, L. and Khanna, P.: A framework for live and cross platform fingerspelling recognition using modified shape matrix variants on depth silhouettes. Computer Vision and Image Understanding vol. 141, Elsevier Ltd., pp. 138–151, 2015

[KoRa14] Kong, W. W. and Ranganath, S.: Towards subject independent continuous sign language recognition: A segment and merge approach. Pattern Recognition vol. 47, Elsevier, Nr. 3, pp. 1294–1308, 2014

[LiKK16] Li, T. H. S., Kao, M., and Kuo, P.: Recognition System for Home-Service-Related Sign Language Using Entropy-Based K -Means Algorithm and ABC-Based HMM. IEEE Transactions on Systems, Man, and Cybernetics: Systems vol. 46, Nr. 1, pp. 150–162, 2016

[MQSA14] Masood, S., Qureshi, M. P., Shah, M. B., Ashraf, S., Halim, Z., and Abbas, G.: Dynamic time wrapping based gesture recognition. 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering, iCREATE 2014 - Proceedings, pp. 205–210, 2014

[OsWy13] Oszust, M. and Wysocki, M.: Polish sign language words recognition with Kinect. 2013 6th International Conference on Human System Interactions, HSI 2013, pp. 219–226, 2013

[RAWD13] Rakun, E., Andriani, M., Wiprayoga, I. W., Danniswara, K., and Tjandra, A.: Combining depth image and skeleton data from Kinect for recognizing words in the sign system for Indonesian language (SIBI [Sistem Isyarat Bahasa Indonesia]). 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 387–392, 2013

[RMPS15] Raheja, J. L., Minhas, M., Prashanth, D., Shah, T., and Chaudhary, A.: Robust gesture recognition using Kinect: A comparison between DTW and HMM. Optik vol. 126, Elsevier GmbH., Nr. 11-12, pp. 1098–1104, 2015

[ThPM14] Theodorakis, S., Pitsikalis, V., and Maragos, P.: Dynamic-static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. Image and Vision Computing vol. 32, Elsevier B.V., Nr. 8, pp. 533–549, 2014

[VeAC13] Verma, H. V., Aggarwal, E., and Chandra, S.: Gesture recognition using kinect for sign language translation. 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), pp. 96–100, 2013

[WCZC15] Wang, H., Chai, X., Zhou, Y., and Chen, X.: Fast sign language recognition benefited from low rank approximation. 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015, 2015

[YaSL10] Yang, R., Sarkar, S., and Loeding, B.: Handling Movement Epenthesis and Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 32, Nr. 3, pp. 462–477, 2010

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Relation of Instant Radiosity Method with Local Estimations of Monte Carlo Method

Victor D. Chembaev
Light Engineering Department Moscow Power Engineering Institute, Moscow, Russia
chembervint@gmail.com

Victor S. Zheltov
Light Engineering Department Moscow Power Engineering Institute, Moscow, Russia
zheltov@list.ru

Vladimir P. Budak,
Light Engineering Department Moscow Power Engineering Institute, Moscow, Russia
budakvp@mpei.ru

Renat S. Notfulin,
Light Engineering Department Moscow Power Engineering Institute, Moscow, Russia
renat@notfullin.com

ABSTRACT

This article discusses mathematical foundations of local estimations of the Monte Carlo method. The basic algorithm of visualization of the 3D scenes based on local estimations, which are an analog of the famous algorithm Instant Radiosity, is considered.

An algorithm for radiance object view-independent calculation based on local estimations of Monte Carlo method is shown

Additionally, questions of representation of radiance object as spherical harmonics expansion in each computational point are analyzed. The assumption of possible direct calculation of radiance object coefficients of expansion in spherical harmonics by Monte Carlo method is brought in, and problems are identified.

Keywords

Radiosity, instant radiosity, global illumination, local estimations, Monte-Carlo, spherical harmonics, view-independent global illumination.

1. INTRODUCTION

Lighting systems simulation and visualization of 3D scenes in computer graphics are based on well-known global lighting equation [Kajiya J. T. 1986].

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{\pi}\int L(\mathbf{r},\hat{\mathbf{l}}')\ \sigma(\mathbf{r};\hat{\mathbf{l}},\hat{\mathbf{l}}')\left|(\hat{\mathbf{N}},\hat{\mathbf{l}}')\right| d\hat{\mathbf{l}}', \quad (1)$$

where $L(\mathbf{r},\hat{\mathbf{l}})$ is the radiance at the point r in the direction $\hat{\mathbf{l}}$, $\sigma(\mathbf{r};\hat{\mathbf{l}},\hat{\mathbf{l}}')$ is the bidirectional scattering distribution function (reflectance or transmittance), L_0 is the radiance of the direct radiation straight near the sources, $\hat{\mathbf{N}}$ is the normal at the point r to the surface of the scene.

The spatial angular distribution of radiance can be calculated on the global illumination ground. It will allow determining light qualitative characteristics (glare, discomfort), which will enable to calculate lighting systems for a specified quality of illumination. The spatial angular radiance distribution calculating algorithm is also the basis for the visualization of 3D scenes. Today, the radiosity is used for the lighting systems simulation. This algorithm is based on the finite element method of radiosity equation. [Goral et al. 1984] [Moon P. 1940].

$$M(\mathbf{r}) = M_0(\mathbf{r}) + \frac{\sigma}{\pi}\int_{\Sigma} M(\mathbf{r}')\Theta(\mathbf{r},\mathbf{r}')\ F(\mathbf{r},\mathbf{r}')\ d^2\mathbf{r}', \quad (2)$$

where $M(\mathbf{r})$ is the radiance at the surface point $\mathbf{r}$, $M_0(\mathbf{r})$ is radiancy at the point $\mathbf{r}$, received straight from the light source, $\Theta(\mathbf{r},\mathbf{r}')$ is the visibility function of an element $d^2\mathbf{r}'$ from point $\mathbf{r}$, $F = \frac{\left|(\hat{\mathbf{N}}(\mathbf{r}),(\mathbf{r}-\mathbf{r}'))\right|\left|(\hat{\mathbf{N}}(\mathbf{r}'),(\mathbf{r}-\mathbf{r}'))\right|}{(\mathbf{r}-\mathbf{r}')^4}$ is the elementary form-factor, $\hat{\mathbf{N}}(\mathbf{r})$ is a normal at the point $\mathbf{r}$ to the scene surface.

Should be noted that the radiosity equation has two analytic solutions. First is a well-known photometric sphere. Second is the Sobolev problem: two parallel infinite diffuse planes and isotropic point source in between [Budak V., Zheltov V. 2014].

Radiosity method on which the lighting systems modeling programs DIALux and Reluxe are based fails to take account of the reflection from non-diffuse surfaces. It markedly affects the determination accuracy of radiance spatial angular distribution.

Recently an interesting algorithm of instant radiosity was introduced [Keller A. 1997], which is a kind of local estimation algorithms of Monte Carlo method [Kalos M. 1963]. However, the phenomenological approach used for derivation makes the algorithm difficult to use in the general case. We undertook the complete proof of strict local assessment algorithms based on the global illumination equation.

In the article, we consider local and double local estimations of the Monte Carlo method for the global illumination equation. Algorithm based on the local estimations allows modeling the radiance of a scene surface point in a given direction. Also, basing on the local estimations, we proposed an algorithm of view-independent determination of the radiance angular distribution on the scene surfaces. The algorithm has obvious advantages over the radiosity method for integrating diffusely directed reflection model.

2. GLOBAL ILLUMINATION EQUATION

From the integral equation for the solid angle, one can go to the well-known integral equation of Fredholm second kind on surfaces

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{\pi}\int_{(\Sigma)} L(\mathbf{r}',\hat{\mathbf{l}}')\ \ (\mathbf{r};\hat{\mathbf{l}}',\hat{\mathbf{l}})F(\mathbf{r}',\mathbf{r})d^2r'\ , \quad (3)$$

where $F(\mathbf{r}',\mathbf{r}) = \dfrac{\left|(\hat{\mathbf{N}}(\mathbf{r}),\mathbf{r}-\mathbf{r}')(\hat{\mathbf{N}}(\mathbf{r}'),\mathbf{r}-\mathbf{r}')\right|}{(\mathbf{r}-\mathbf{r}')^4}\Theta(\mathbf{r}',\mathbf{r})$,

$\hat{\mathbf{l}}' = \dfrac{\mathbf{r}-\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|}$.

One can construct an algorithm based on (3) for its solution by Monte Carlo method. However, wandering along the surfaces Σ of the scene visualization is not a trivial task. Conventional scheme of wandering of the Monte Carlo method is constructed in space, which requires the integral to integration over the volume.

Integral over the volume

$$d^3r' = |\mathbf{r}-\mathbf{r}'|^2\, dr' d\hat{\mathbf{l}}',$$

$$d\hat{\mathbf{l}}' = \frac{\left|(\hat{\mathbf{N}}(\mathbf{r}'),\mathbf{r}-\mathbf{r}')\right|}{(\mathbf{r}-\mathbf{r}')^2}d^2r'. \quad (4)$$

For integration over dr' we will use equivalent transformation with usage δ-function properties

$$\int_{(\Sigma)} L(\mathbf{r}',\hat{\mathbf{l}}')\sigma(\mathbf{r};\hat{\mathbf{l}}',\hat{\mathbf{l}})F(\mathbf{r}',\mathbf{r})d^2r' \equiv$$

$$\equiv \int_0^\infty\int L(\mathbf{r}',\hat{\mathbf{l}}')\sigma(\mathbf{r};\hat{\mathbf{l}}',\hat{\mathbf{l}})\frac{F(\mathbf{r}',\mathbf{r})}{\left|(\hat{\mathbf{N}}(\mathbf{r}'),\hat{\mathbf{l}}')\right|}|\mathbf{r}-\mathbf{r}'|^2 \times$$

$$\times\frac{\left|(\hat{\mathbf{N}}(\mathbf{r}'),\hat{\mathbf{l}}')\right|d^2r'}{|\mathbf{r}-\mathbf{r}'|^2}\delta(\xi_0-|\mathbf{r}-\mathbf{r}'|)dr', \quad (5)$$

where ξ_0 is a solution of the surface Σ equation $\Pi(\mathbf{r})$=0: $\Pi(\mathbf{r}-\xi_0\hat{\mathbf{l}}')=0$.

The surface equation can be included directly in (5) because the ratios

$$\xi_0 - |\mathbf{r}-\mathbf{r}'| = 0 \text{ and } \Pi(\mathbf{r}-|\mathbf{r}-\mathbf{r}'|\hat{\mathbf{l}}') = 0 \quad (6)$$

are equivalent.

At that, it is important to consider the δ -function properties.

$$\int_a^b \delta(f(x))dx = \frac{1}{\left|\dfrac{df(x)}{dx}\right|_{x=x_0}}\int_a^b \delta(x-x_0)dx, f(x_0) = 0. \quad (7)$$

Accordingly, we will get for global illumination equation

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{\pi}\int L(\mathbf{r}',\hat{\mathbf{l}}')\ \ (\mathbf{r};\hat{\mathbf{l}}',\hat{\mathbf{l}})G(\mathbf{r}',\mathbf{r})d^3r' \quad (8)$$

where the new geometric factor

$$G(\mathbf{r}',\mathbf{r}) = \frac{\left|(\hat{\mathbf{N}}(\mathbf{r}),\mathbf{r}-\mathbf{r}')\right|}{(\mathbf{r}-\mathbf{r}')^3}\Theta(\mathbf{r}',\mathbf{r})\times$$

$$\times\left|\frac{d\Pi(\mathbf{r}-\xi\hat{\mathbf{l}}')}{d\xi}\right|_{\xi=|\mathbf{r}-\mathbf{r}_0|}\delta\left(\Pi(\mathbf{r}-|\mathbf{r}-\mathbf{r}'|\hat{\mathbf{l}}')\right), \quad (9)$$

where $\Pi(\mathbf{r}-|\mathbf{r}-\mathbf{r}_0|\hat{\mathbf{l}}') = 0,\ \hat{\mathbf{l}}' = \dfrac{\mathbf{r}-\mathbf{r}_0}{|\mathbf{r}-\mathbf{r}_0|}$.

Should be noted that equation (8) is derived for the radiance of a point on the surface Σ. However, light qualitative characteristics (glare, discomfort) are indissolubly related to observer: the radiance should be determined by some point in space. Formally, for the 3D visualization the radiance on the camera in space is also determined. Let us consider the equation about an arbitrary point in space.

The radiance angular distribution $L_\Sigma(\mathbf{r},\hat{\mathbf{l}})$ on a closed surface Σ is defined by the equation Π(r) = 0. It is required to determine the distribution of radiance in an arbitrary point **r** in volume *V* limited by the surface Σ. The volume is filled with a completely transparent medium.

By the solution of the radiative transfer equation for a transparent medium, radiance along the ray does not change. Therefore, the radiance of the point **r** in the direction $\hat{\mathbf{l}}$ is equal to the surface at the point of intersection of the surface with a ray from a point **r** at the direction $\hat{\mathbf{l}}$:

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_{\Sigma}(\mathbf{r} - \xi\hat{\mathbf{l}},\hat{\mathbf{l}}), \quad (10)$$

where ξ – root of the equation

$$\Pi(\mathbf{r} - \xi\hat{\mathbf{l}}) = 0 \quad (11)$$

The last correlations can be made user-friendlier analytically when properties of δ-function are used:

$$L(\mathbf{r},\hat{\mathbf{l}}) = C_{01}\int_{(V')} L_{\Sigma}(\mathbf{r}',\hat{\mathbf{l}})\delta(\Pi(\mathbf{r} - |\mathbf{r} - \mathbf{r}'|\hat{\mathbf{l}})) \times$$

$$\times\delta(\hat{\mathbf{l}} - \hat{\mathbf{l}}_0)\frac{d^3r'}{(\mathbf{r} - \mathbf{r}')^2}, \quad (12)$$

where $C_{01} = \left|\frac{d\Pi(\mathbf{r} - \xi\hat{\mathbf{l}})}{d\xi}\right|_{\xi = |\mathbf{r} - \mathbf{r}_0|}$ is related to the properties of the integral of δ-function with a composite argument, $\hat{\mathbf{l}}_0 = \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|}$.

Combining the above expression for scene surface radiance (8) and radiance for a point in space (12), we can write the final expression

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_{\Sigma 0}(\mathbf{r}_{\Sigma},\hat{\mathbf{l}}) +$$

$$+\frac{1}{\pi}C_{01}\int L_{\Sigma}(\mathbf{r}_1,\hat{\mathbf{l}}')\sigma(\mathbf{r}_2;\hat{\mathbf{l}}',\hat{\mathbf{l}})G(\mathbf{r}_1,\mathbf{r}_2) \times$$

$$\times\delta(\Pi(\mathbf{r} - |\mathbf{r} - \mathbf{r}_2|\hat{\mathbf{l}}))d^3r_1 d^3r_2, \quad (13)$$

where point $\mathbf{r}_{\Sigma}$ corresponds to the point of intersection of the camera sight line with surface Σ.

Thus, the last equation describes the radiance in any point of space.

3. LOCAL ESTIMATIONS

Local estimates were formulated in atomic physics [Kalos M.H. 1963] and continued its development in the optics of the atmosphere and ocean when solving the radiation transport equation [Marchuk G.I. 1980]. Note that global illumination equation is an implication of the radiative transfer equation in a vacuum. Let's consider local estimations for global illumination equation.

Local Estimation

The solution (8) can be shown as Neumann series

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{\pi}\int L_0(\mathbf{r}_1,\hat{\mathbf{l}}_1)\sigma(\mathbf{r};\hat{\mathbf{l}}_1,\hat{\mathbf{l}})G(\mathbf{r}_1,\mathbf{r})d^3r_1$$

$$+\frac{1}{\pi}\int\frac{1}{\pi}\int L_0(\mathbf{r}_1,\hat{\mathbf{l}}_1)\sigma(\mathbf{r}_2;\hat{\mathbf{l}}_1,\hat{\mathbf{l}}_2)G(\mathbf{r}_1,\mathbf{r}_2)d^3r_1\sigma(\mathbf{r};\hat{\mathbf{l}}_2,\hat{\mathbf{l}}) \times$$

$$\times G(\mathbf{r}_2,\mathbf{r})d^3r_1 + \quad (14)$$

All terms of the series - definite integrals, which will be calculated by the Monte Carlo

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{\pi}\frac{1}{N}\sum_{i=1}^{N}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}})G(\mathbf{r}_1,\mathbf{r})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r},\hat{\mathbf{l}})} +$$

$$+\frac{1}{\pi^2}\frac{1}{N}\sum_{i=1}^{N}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r}_{2i};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}}_{2i})G(\mathbf{r}_{1i},\mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i})} \times$$

$$\times\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{2i},\hat{\mathbf{l}})G(\mathbf{r}_{2i},\mathbf{r})}{p_2(\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i}\to\mathbf{r},\hat{\mathbf{l}})} + \quad (15)$$

Combining the sums into one

$$L(\mathbf{r},\hat{\mathbf{l}}) = L_0(\mathbf{r},\hat{\mathbf{l}}) + \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{\pi}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}})G(\mathbf{r}_1,\mathbf{r})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r},\hat{\mathbf{l}})}\right.$$

$$+\frac{1}{\pi^2}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r}_{2i};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}}_{2i})G(\mathbf{r}_{1i},\mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i})} \times$$

$$\left.\times\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{2i},\hat{\mathbf{l}})G(\mathbf{r}_{2i},\mathbf{r})}{p_2(\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i}\to\mathbf{r},\hat{\mathbf{l}})} + \right) \quad (16)$$

The last expression can be interpreted as a Markov chain wandering ray with the contribution by the kernel

$$k(x_i \to x) = \frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_i,\hat{\mathbf{l}})G(\mathbf{r}_i,\mathbf{r})}{p_2(x_i \to x)}. \quad (17)$$

Similar expressions were presented in [Budak et al. 2015.] As a result of the construction of the Markov chain, we can evaluate the radiance at a given point in a given direction on the scene surface. Such estimation may be called local estimation of Monte Carlo. Similarly to the estimation, made for the transport equation in atmospheric optics [Marchuk G.I. 1980].

Thus, local estimation allows calculating surface radiance at a given scene point in a given direction.

Let's consider the algorithm for radiance calculation with the local estimates of Monte Carlo. Accept that we have some 3D scene. We fix points and the directions on the surface at which we want to determine the radiance.

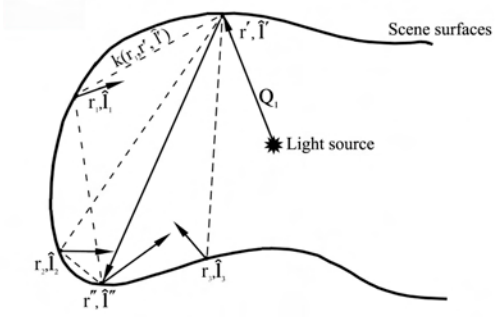


Figure 2: Algorithm scheme of radiance evaluation by local estimation

Let us cast the ray from the source. The most efficient way to select a ray is an importance sampling, but any other known samples for Monte Carlo methods can be used. This ray will receive the weight corresponding to the radiance. Determine the point of intersection of

the ray with a scene element. Then we can evaluate the equation (8) kernel (17) for each of the test points and calculate directly reflected radiance in the test point taking into account its reflection coefficient in the test direction. Then, casting a new ray, considering reflectance coefficient. Its weight decreases. The process continues iteratively until the ray's weight is below the threshold or until it leaves the scene. Then again we take a new ray from the source. When statistics is accumulated, averaged and normalized, we will get the radiance directly at predetermined points in a given direction. Also, should be noted that the algorithm can be implemented in "Russian roulette" principle. Method is outlined in Figure 2.

Double Local Estimation

Let's take a look at the local estimation algorithm construction to determine the radiance of a given point in space. In equation (13) appears an additional δ-function $\delta(\Pi(\mathbf{r}-|\mathbf{r}-\mathbf{r}_2|\hat{\mathbf{l}}))$ which depends on the direction $\hat{\mathbf{l}}$. It makes direct modeling impossible. It is clearly seen in the graphic interpretation in Figure 3.

Suppose we have a given point $\mathbf{r}$ in space and a direction $\hat{\mathbf{l}}$ in which we want to determine the radiance. We begin to build a Markov chain. As it seen in Figure 3, it is impossible to get from the chain point in test direction at the test point. To solve the problem we fix an additional node - the point on the surface - and do calculations through it. This approach is called a double local estimation [Marchuk GI 1980].

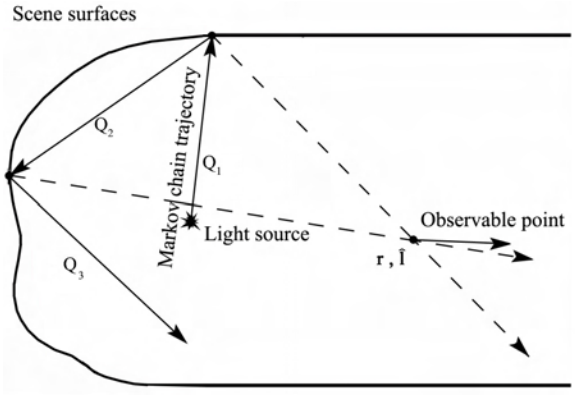


Figure 3: Geometric description of the impossibility of radiance modeling direct in the spatial point

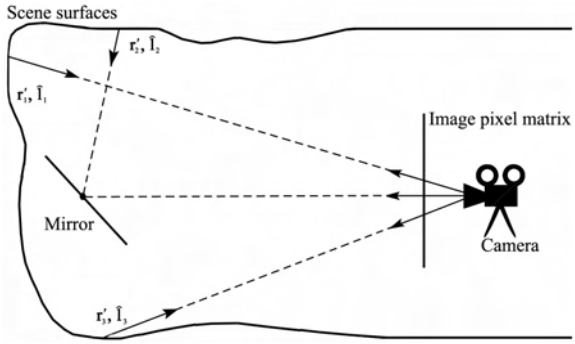


Figure 4: Scheme of scenes three-dimensional visualization construction

The further algorithm will be no different from the local estimation.

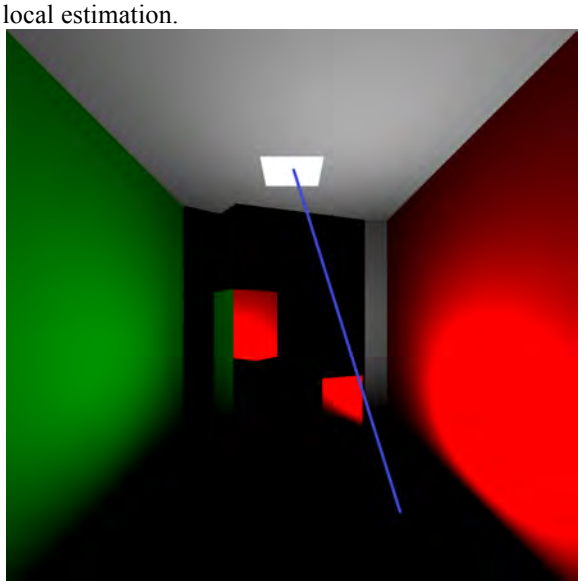


Figure 5: Scene rendering by one ray on one node of Markov chain

On Figure 5 presented the rendering of the 3D scene on one node of the Markov chain. In fact, even on one node, we get all the images at once, taking into account multiple reflections. Certainly, it is not accurate. The figure shows an image of the scene Figure 6 considering a second node of the Markov chain for the same ray. After drawing a large number of rays, we can get the final image shown in Figure 7.

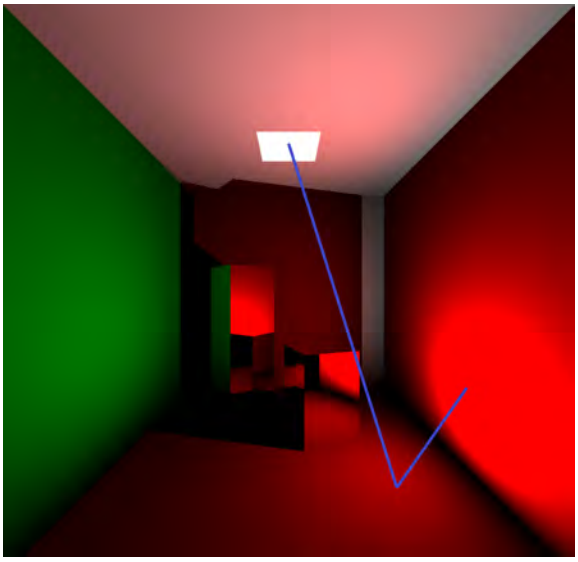


Figure 6: Scene rendering by one ray on two nodes of Markov chain

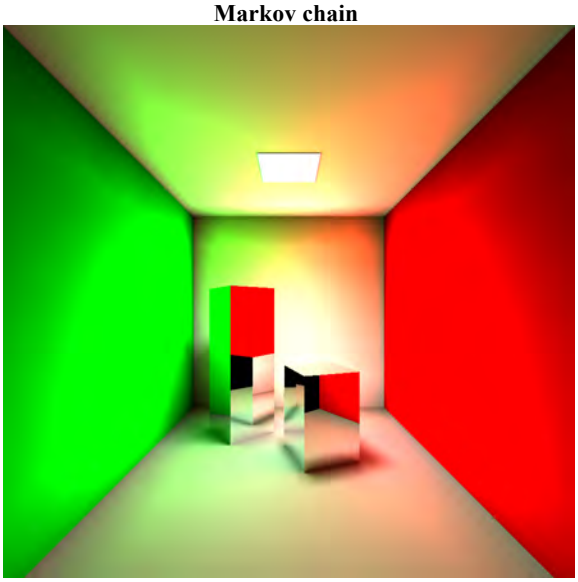


Figure 7: Scene rendering by 1000 rays on average five nodes of Markov chain

Local Estimations and Instant Radiosity

Local estimations of Monte Carlo were proposed for the first time in phenomenological approach in the work of [Keller A. 1997] and were called Instant Radiosity.

Should be noted that the algorithm described in [Keller A. 1997] is different from the one proposed in this article and based on the local estimations. The author divides the process into two stages - forming the virtual light sources and a calculation of their contribution. From our point of view, the construction of Markov chains and calculation of its nodes contribution are closely interrelated. Should be noted that analysis of instant radiosity method in [Pharr M., Humphreys G. 2010] is based on a similar processes separation approach.

In the approach wet put forth in our work, these processes are not divided. Because of that, we can see the whole image at any time. In other words, we can get a complete image at once even by one ray of Markov chain.

The geometric factor kernel (17) of global illumination contains a well-known feature $\frac{1}{(\mathbf{r}-\mathbf{r}')^3}$ that leads to local estimation infinite dispersion [Kollig. T., Keller A. 2004]. There are two algorithms for its elimination: with proposed equation kernel restriction and further solutions refinement by integration within this volume; or with integration by a small area around the observation point in which averaging of results will take place [Kalos MH 1963].

4. VIEW-INDEPENDENT LOCAL ESTIMATION

Radiosity method is not widespread in computer graphics. Nevertheless, it found its application in lighting design systems.

Currently, the radiosity method is used in two main software products for lighting systems design, DIAlux and Relux. The main advantage of the method is that it allows calculating of global illumination without camera position - a view-independent calculation. However, it uses a diffuse reflection model, which cannot describe real materials. Moreover, the calculated illumination distribution is not a characteristic perceived by the eye.

Based on the described local estimations of Monte Carlo method, we can build a new algorithm for calculating global illumination without camera position and with any reflection model.

Algorithm

As in the case of radiosity, the calculations will be made on the mesh. This mesh can be both simple static, formed before the calculations, based on simple criteria, or dynamically generated directly during the computation. Should be noted that according to the circumstantial evidence we assume that DIAlux uses static mesh. At the same time, our mesh will differ significantly from one used in radiosity method. In the radiosity method, the radiance is averaged by the mesh element. In our case, we can directly calculate the radiance at a given point in a fixed direction and can perform calculations on the mesh nodes.

For simplicity, we will use a static mesh. Suppose we have some initial scene. We divide it into smaller mesh nodes and define a uniform step zenith θ and azimuthal angle φ direction by the normally oriented hemisphere. These are directions at fixed points in which we calculate the radiance $L(\mathbf{r},\hat{\mathbf{l}})$.

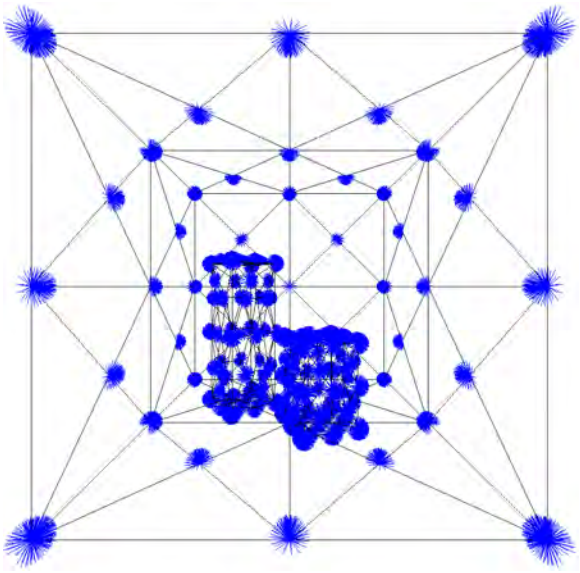


Figure 9: Definition of calculation points and directions of the view-independent calculation

The further algorithm will be no different from the local estimation discussed earlier. After calculation using the local estimation, we will get the radiance values at the mesh nodes in the set of directions.

To draw an analogy with radiosity method, further, when doing final image collection or analyzing illumination in lighting calculations, we will be interested in the radiance of arbitrary points on the surface. However, while the radiance value in radiosity method does not depend on the position of the viewpoint (camera), in our case the radiance will depend on it.

To determine the radiance at any point of the element, we need first to find the radiance at the vertexes of a triangular mesh element in the direction of the observer. It can be done by approximating the radiance calculated in directions distributed over a hemisphere. Then we can calculate the radiance at a given point within the triangular element through barycentric coordinates.

Obviously, the accuracy will directly depend on the size of scene partition element and the number of directions in which the radiance will be calculated.

The problem of choosing the number of directions for calculations is beyond the scope of this study. However, our preliminary studies show that, for example, to the Phong model in real scenes, the higher the degree of the cosine, the greater the number of directions necessary to describe that.

Figures 10 and 11 shows an example of view independent calculation of the Sobolev problem scene with rectangle downlight. The scene surfaces have Phong reflectance with the power of cosine equals 16.

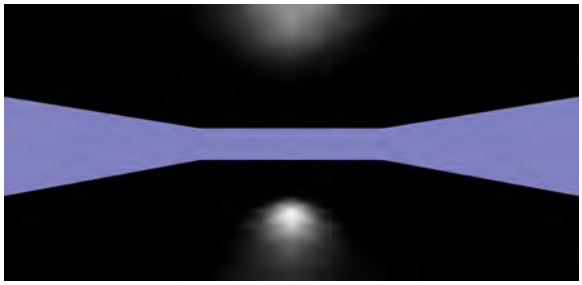


Figure 10: The Sobolev problem scene view-independent rendering

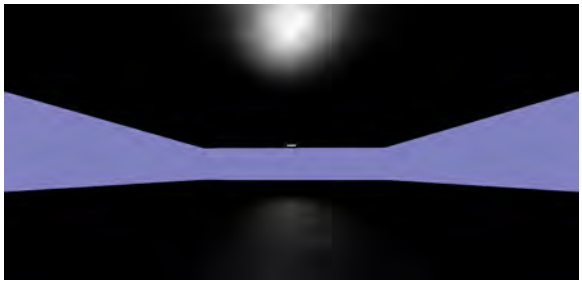


Figure 11: The Sobolev problem scene view-independent rendering

Using spherical harmonics approximation

As described above, when analyzing the obtained results we have a problem of radiance approximating at a point in the direction. We suggest a well-known expansion by spherical functions for this. To do so, we can expand the radiance at each point by spherical harmonics

$$L(\mathbf{r},\hat{\mathbf{l}}) = \sum_{n=0}^{N}\sum_{m=-n}^{n} C_n^m(\mathbf{r})\,\mathrm{Y}_n^m(\hat{\mathbf{l}}) =$$

$$= \sum_{n=0}^{N}\sum_{m=0}^{n}\left(A_n^m(\mathbf{r})\cos\varphi + B_n^m(\mathbf{r})\sin\varphi\right)\mathrm{P}_n^m(\hat{\mathbf{l}}\cdot\hat{\mathbf{z}}), \quad (18)$$

where

$$C_n^m(\mathbf{r}) = \int \mathrm{Y}_n^m(\hat{\mathbf{l}})L(\mathbf{r},\hat{\mathbf{l}})d\hat{\mathbf{l}}\,. \quad (19)$$

As a result, for each vertex of the mesh after the calculation, we will store expansion coefficients $A_n^m(\mathbf{r})$ and $B_n^m(\mathbf{r})$ instead of radiance by the set of generated directions $L(\mathbf{r},\hat{\mathbf{l}})$. Therefore, we can also determine the radiance in the desired direction $\hat{\mathbf{l}}$ based on these coefficients. According to our preliminary study, this can significantly reduce the amount of stored information.

Also, should be noted that for the greater efficiency of the algorithm of directions at the zenith angle θ in the formation of directions mesh at the point it is better to choose in zeros of the Legendre polynomials. It will further on when determining expansion coefficients by integrating allow using the Gaussian quadrature, which gives the exact value by integration.

An important fact is that the loss of "energy" does not depend on spherical harmonics series terms number. Each next following term clarifies the solution.

Radiance object expansion by spherical harmonics also essential from the lighting technology science perspective, because we can see that some members of the series have a rigorous physical interpretation, for example, the coefficient $A_0^0(\mathbf{r})$ will be the same as scalar irradiance - the radiance integral over the solid angle since $Y_0^0(\theta,\varphi)= 1$. Coefficient $A_1^0(\mathbf{r})$ - as light vector module because $Y_1^0(\theta,\varphi) = \cos$.

Way to increase Spherical Harmonics Transformation (SHT) algorithm performance

Reducing the amount of information stored in the brightness distribution at the points of the scene is can be achieved by introduction of an additional algorithm (SHT), which certainly makes a negative contribution to the performance of the whole algorithm. Taking into account the actual for today resolutions, it becomes obvious that the cyclical structure of SHT procedure imposes the requirement for the high performance of this algorithm.

It is clear from the definition that integration by volume is required to expand a function in a series of spherical harmonics. However, one may notice that the integral over the azimuthal angle φ (the sum over **n** in (19)) is the Fourier series. It is known that there is an effective numerical method, called the Fast Fourier Transform (FFT) for the Fourier transform procedure. Thus, from the algorithmic point of view, the double integral is reduced to the single on ϑ, and inner integral can be calculated using the FFT [Martin J. Mohlenkamp 1999].

One property of associated Legendre polynomials $Y_n^m(\cos\varphi)$ is that it is either even or odd across φ=π/2 as *n-m* is even or odd. The use of equalities also reduces computation time by a factor of two [Martin J. Mohlenkamp 1999].

Other important factors affecting performance are the calculation of $Y_n^m(\cos\varphi)$and the step of sampling, which will affect the speed of calculation of the integral. Taking into account specific of our task iterative calculation of the associated Legendre polynomials seems expensive. However, we guess that any radiance object should fit well into the only single matrix of angles sampling, which enables us to calculate polynomials in advance. We guess that sampling rate 2n should be sufficient.

Spherical Harmonics Local Estimation

One of the problems of radiosity method is a rather large consumption of memory for storing the information on the mesh. In our method of View-Independent local estimation, this amount also increases by some directions for each node. As described above when using radiance decomposition by spherical harmonics we can reduce the amount of stored information. Thus, if we look for a solution directly in the spherical functions, we can immediately calculate the expansion coefficients for the given points. In this case, the expression (20) can be directly estimated by Monte-Carlo method already on one node, and then (15) can be written as

$$C_n^m(\mathbf{r}) = \frac{1}{N}\sum_i L_0(\mathbf{r},\hat{\mathbf{l}})\,\mathrm{Y}_n^m(\hat{\mathbf{l}}) +$$

$$+\frac{1}{\pi}\frac{1}{N}\sum_{i=1}^{N}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}})G(\mathbf{r}_1,\mathbf{r})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r},\hat{\mathbf{l}})}\mathrm{Y}_n^m(\hat{\mathbf{l}}) +$$

$$+\frac{1}{\pi^2}\frac{1}{N}\sum_{i=1}^{N}\frac{L_0(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i})}\frac{\sigma(\mathbf{r}_{2i};\hat{\mathbf{l}}_{1i},\hat{\mathbf{l}}_{2i})G(\mathbf{r}_{1i},\mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i},\hat{\mathbf{l}}_{1i}\to\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i})} +$$

$$+\frac{\sigma(\mathbf{r};\hat{\mathbf{l}}_{2i},\hat{\mathbf{l}})G(\mathbf{r}_{2i},\mathbf{r})}{p_2(\mathbf{r}_{2i},\hat{\mathbf{l}}_{2i}\to\mathbf{r},\hat{\mathbf{l}})}\mathrm{Y}_n^m(\hat{\mathbf{l}}) + \qquad (20)$$

Thus, in the spherical harmonics local estimation algorithm we do not fix the directions at the vertexes of the mesh, but on each node of the Markov chain, we determine a random direction of reflection for each vertex and calculate the expansion coefficients with one radiance value. The statistics are collected directly in the expansion coefficients $A_n^m(\mathbf{r})$ and $B_n^m(\mathbf{r})$.

5. CONCLUSION AND FUTURE WORK

Local estimation of the Monte Carlo method allows calculating the radiance of a given point on the surface or in space in a particular direction. Obtained expressions show us the connection between Instant Radiosity method and local estimation, harmonically complementing the already known method.

Local estimations allow obtaining the spatially angular distribution of radiance, and view-independent simulation algorithm of allocation allows to get to a new level of illumination quality analysis. Avoiding reflections diffuse model used in lighting simulation nowadays is a necessary stage in the transition from designing of lighting systems with defined quantitative characteristics to the simulation of lighting systems with specified quality characteristics.

The work still has numerous unsolved issues. In the future, our attention will be paid to:

- distinctive features in the core of the global illumination equation;
- finding a solution directly in spherical functions decomposition coefficients;

performance issues and as the ultimate goal - the quality of lighting.

6. ACKNOWLEDGMENTS

The Ministry of Education and Science of the Russian Federation (Project No 2487) supported this work.

7. REFERENCES

[Goral et al. 1984] GORAL, C., TORRANCE, K.E., GREENBERG, D.P., BATTAILE, B. Modeling the interaction of light between diffuse surfaces. Computer Graphics, Vol. 18, No. 3, 1984.

[Kajiya J. T. 1986] KAJIYA, J.T. The rendering equation. In Proceedings of SIGGRAPH 1986. V.20, N4. – P.143-150, 1986.

[Moon P. 1940] MOON, P. On Interreflections. JOSA, 1940 Vol. 30. N2. P. 195 –205, 1940.

[Budak V., Zheltov V. 2014]. BUDAK, V., ZHELTOV, V. Local Monte Carlo estimation methods in the solution of global illumination equation. In WSCG 2014 Communication Papers Proceedings. P. 25-31, 2014.

[Keller A. 1997] GORAL, C., TORRANCE, K.E., GREENBERG, D.P., BATTAILE, B. Modeling the interaction of light between diffuse surfaces. Computer Graphics, Vol. 18, No. 3, 1984.

[Kalos M. 1963] KALOS, M.H. On the Estimation of Flux at a Point by Monte Carlo. Nuclear Science and Engineering. 1963, Vol. 16, N.1, p.111-117, 1963.

[Marchuk G.I. 1980] MARCHUK, G.I. Monte-Carlo Methods in Atmospheric Optics. Berlin: Springer-Verlag, 1980.

[Budak et al. 2015] CHEMBAEV, V.D., BUDAK, V.P., ZHELTOV, V.S., NOTFULLIN, R.S., SELIVANOV, V.A. Usage of local estimations at the solution of global illumination equation. In Proceedings of GRAPHICON 2015, 7–11, 2015.

[Pharr M., Humphreys G. 2010] PHARR, M., HUMPHREYS, G. Physically Based Rendering, 2nd Edition, From Theory to Implementation. Morgan Kaufmann, 2010.

[Kollig. T., Keller A. 2004] KOLLIG, T., KELLER, A. Illumination in the presence of weak singularities. In Proceedings of Monte Carlo and Quasi-Monte Carlo Methods, 2004.

[Martin J. Mohlenkamp 1999] Martin, J. Mohlenkamp. The Journal of Fourier Analysis and Applications 5(2/3):159-184, 1999.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

JIT-Compilation for Interactive Scientific Visualization

J. S. Mueller-Roemer
Fraunhofer IGD, TU Darmstadt
Fraunhoferstr. 5
Germany 64283, Darmstadt, Hessen
Johannes.Mueller-Roemer@igd.fraunhofer.de

C. Altenhofen
Fraunhofer IGD, TU Darmstadt
Fraunhoferstr. 5
Germany 64283, Darmstadt, Hessen
Christian.Altenhofen@igd.fraunhofer.de

ABSTRACT

Due to the proliferation of mobile devices and cloud computing, remote simulation and visualization have become increasingly important. In order to reduce bandwidth and (de)serialization costs, and to improve mobile battery life, we examine the performance and bandwidth benefits of using an optimizing query compiler for remote post-processing of interactive and in-situ simulations. We conduct a detailed analysis of streaming performance for interactive simulations. By evaluating pre-compiled expressions and only sending one calculated field instead of the raw simulation results, we reduce the amount of data transmitted over the network by up to $2/3$ for our test cases. A CPU and a GPU version of the query compiler are implemented and evaluated. The latter is used to additionally reduce PCIe bus bandwidth costs and provides an improvement of over 70% relative to the CPU implementation when using a GPU-based simulation back-end.

Keywords

Scientific Visualization, Network graphics, Mobile Computing, Compilers, LLVM, JIT

1 INTRODUCTION

In modern computer-aided engineering (CAE), compute-intensive simulations are more and more often run on remote cloud or high-performance computing (HPC) infrastructures. To avoid downloading large simulation results to a local client machine, solutions for remote visualization and remote post-processing are needed. Although the option of using a standard visualization tool via a video streaming system such as Virtual Network Computing (VNC) [Ric+98] is attractive, it is desirable to keep latencies to a minimum to increase usability [TAS06]. By transferring (partial) floating point simulation data instead, operations such as probing or changes in color mapping can be performed locally with minimal latency. Similarly, by transferring geometry or point data in 3D, smooth camera interaction becomes possible [Alt+16].

In particular, we aim to answer the following questions:

1. Can compiler technologies be used to decrease visualization latencies in a remote scientific visualization system?

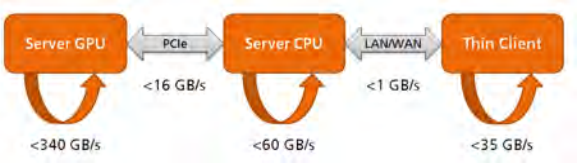


Figure 1: Network, bus and memory bandwidths relevant to streaming a GPU-based simulation. The two most limiting factors are the network bandwidth and the PCIe bus bandwidth.

2. Can GPU-based simulations running at interactive rates profit from GPU-based query compilation?

When individual result fields of a simulation are visualized, data can simply be streamed from the server running the simulation. When viewing derived values that depend on multiple fields such as the total energy density $\frac{v^2}{2} + gz + \frac{p}{\rho}$ in an Eulerian computational fluid dynamics (CFD) simulation, a different solution is required, as transferring all data would be prohibitive, especially when considering comparatively slow mobile connections (see Fig. 1) and mobile power consumption.

A simulation service could provide a fixed set of derived values. However, the derived values a user wants to visualize often depend not only on the physics domain, but also on the application domain. Therefore, compiling such a fixed set requires domain knowledge and is very likely to be incomplete and insufficient for the user to perform his or her work. For stationary simulations, a server-side interpreter for user queries is en-

tirely sufficient, as each query only has to be processed once. For interactive simulations, i.e., time-dependent simulations running at several frames per second, or the in-situ visualization of a long-running solver, however, this approach becomes costly due to the repeated interpretation overhead.

To avoid these costs, we examine the performance and bandwidth benefits of using optimizing compiler technologies for remote, in-situ post-processing and visualization of simulations running at interactive rates. The implemented query compiler has a native CPU back-end (x86 and x86-64) as well as a GPU back-end (NVIDIA PTX). The latter is used to extend the bandwidth savings to the PCIe (Peripheral Component Interconnect Express) bus in addition to the network interface, further improving performance when using GPU-based simulation algorithms. Our approach is easily extended to all platforms supported by LLVM [LA04].

2 RELATED WORK

This section describes existing methods that are related to our approach and briefly shows their benefits and drawbacks.

2.1 Compiler Technologies for Visualization

Previous applications of compilers and domain-specific languages (DSLs) to scientific visualization mostly center on volume visualization and rendering itself [Chi+12; Cho+14; Rau+14]. These systems therefore represent the entire visualization pipeline. In the streaming architecture presented in this paper, data is transformed on the server and rendered on the client. Therefore, the aforementioned systems are not directly applicable. This split corresponds to the two stages "Data Management" and "Picture Synthesis" in the system architecture used by [Duk+09]. However, they use an embedded DSL (eDSL) based on Haskell [Pey03]. As client code must be considered untrusted by the server, a general-purpose language and any eDSL based on such a language pose a great security risk. In the area of visual analytics, MapD Technologies [Map16] have recently used LLVM/NVVM [NVI16] and GPU computing with great success [MŞ15]. In contrast, we aim to bring the advantages of using compiler technologies to the field of scientific visualization, with a focus on interactively changing datasets from either in-situ or interactive simulations.

2.2 Compression

Another approach to reduce bandwidth requirements is to apply floating-point data compression. For structured data, lossy methods such as the one presented in [Lin14] achieve good results. Structured data occurs in a significant subset of simulation domains and such a method would be widely applicable. However, lossy compression before calculation of desired derived values can lead to larger errors in the compounded result. For general data, a method such as the one presented in [OB11] could be used. Their method is a lossless compression method and implemented on the GPU, making it applicable to reducing network as well as PCIe bus bandwidths and to arbitrary simulation domains. As compression is orthogonal to the method presented in this paper, any suitable compression algorithm can be chosen and combined with our approach. However, all compression methods incur an additional computation cost. A good overview of existing compression techniques for floating-point data is given in [RKB06], showing compression ratios as well as compression and decompression times.

2.3 Application Sharing

Although we present a method to reduce the amount of data transferred when the client performs part of the necessary calculations to reduce perceived latency, it is worth mentioning that transmitting the content of single applications or the entire desktop as an image or video stream is still a common way to visualize server applications on (thin) client machines across a local network or the Internet. Microsoft's Remote Desktop Protocol (RDP) [Mic16] or the platform-independent Virtual Network Computing (VNC) [Ric+98] are two popular implementations of this concept. Good results have also been achieved in the area of video streaming for games [Che+11]. However, mobile networks, especially 3G networks, can add several hundreds of milliseconds of latency [Gri13].

As shown in this section, many approaches for remote visualization exist in the context of scientific visualization and visual analytics. However, the potential of compiler technology in the field of remote visualization of interactive simulations has not been discussed yet. Especially in modern high performance computing (HPC) or cloud environments, these techniques can greatly improve usability by optimizing data transmission and increasing update rates on the clients, while minimizing server overhead and latency. Existing compression algorithms can be applied independently to decrease the required bandwidth even further. However, the resulting increase in encoding and decoding time has to be kept in mind.

3 CONCEPT AND IMPLEMENTATION

In this section, we present our prototype visualization system, which consists of:

1. an interactive simulation back-end running on the server

2. a visualization front-end running on the client
3. an application-specific streaming protocol
4. the query expression compiler

Using the streaming protocol, simulation data is transmitted at interactive rates from the server to the client. By transmitting data instead of images, many interactions, for example color map changes, become possible on the client without incurring network round trip and transmission latency. When the user wants to visualize values that are not a direct output of the simulation back-end, the query expression compiler is used to efficiently transform data on the server, reducing network bandwidth requirements. The prototype is based on a CFD simulation back-end, however, the method is directly applicable to other physical domains such as computational solid mechanics (CSM), computational aero-acoustics or computational electrodynamics. For easy reuse with other simulation back-ends, the query compiler is designed as a shared library with a simple interface.

In the following, we briefly outline the simulation back-end as well as the visualization front-end and detail the streaming protocol as well as the query compiler.

3.1 Simulation Back-End

Our query-based streaming prototype is based on an interactive, Eulerian 2D/3D-CFD code for staggered regular grids using a multigrid solver based on the one presented in [Web+15]. All computational kernels are implemented in CUDA [Nic+08]. Therefore, GPU-CPU transfers are only required for data that is sent to the client.

3.2 Visualization Front-End

Two streaming clients have been implemented:

1. A graphical client running on a desktop machine shown in Figure 2.
2. An HTML5+JavaScript client for streaming performance measurements shown in Figure 3.

The former allows user interaction such as selecting the results to show, or entering an expression combining multiple result fields. Furthermore, the color mapping can be interactively modified by manipulating the color ramp widget with the mouse. The latter was developed to determine feasibility of a web client by evaluating streaming performance including deserialization. Both can be used to stream regular 2D and 3D grids. However, visualization is limited to 2D slices in the prototype.

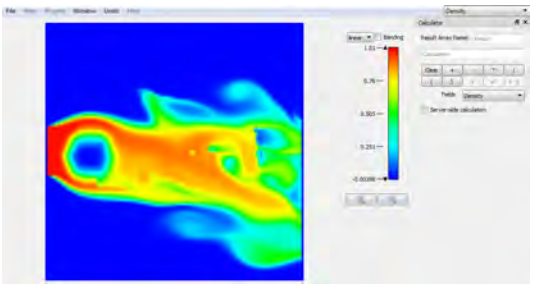


Figure 2: The graphical streaming client. The user can choose which result field to view or enter an expression combining multiple fields. Color mapping can be modified interactively by clicking and dragging the color ramp widget.

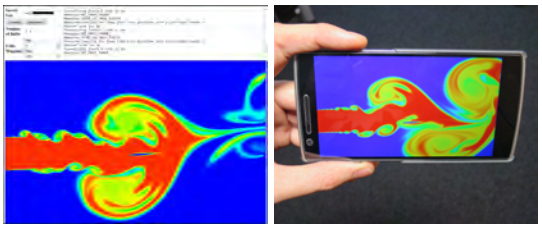


Figure 3: The web-based streaming client can be run in a web browser on desktop computers or mobile devices without installing additional software and provides a simple user interface including 2D visualization and basic logging functionality.

3.3 Streaming Protocol

The streaming protocol is based on Protocol Buffers (ProtoBuf) [Goo08] for serialization and deserialization. ProtoBuf is a platform-independent open source framework that generates serialization and deserialization code from a declarative message description, which greatly simplifies modifications to the protocol. Implementations of ProtoBuf are available for a large number of programming languages, including C++ (as used by our server) and JavaScript. To minimize overhead, the fields of physical values are marked as packed repeated fields, as shown in Listing 1. This prevents ProtoBuf from inserting type tags between each value and ensures that values are transmitted contiguously. The generated messages are transmitted using the WebSocket protocol.

Although WebSockets are based on TCP and have a greater overhead than using UDP, they have several advantages. First, WebSockets ensure that message order is preserved and that all messages are received unless the connection is lost entirely, simplifying client and server implementation. Second, an increasing number of mobile applications are provided as HTML5 web applications and WebSockets are supported by all current browsers, while TCP and UDP are not accessible from JavaScript. This ensures portability of our streaming solution to HTML5+JavaScript.

Listing 1: Main streaming message definition (ProtoBuf [Goo08]), showing the use of a packed repeated field for physical values to reduce overhead.

```
message PostGridFields
{
    required Header header = 1;

    required int32 gridSizeX = 2;
    required int32 gridSizeY = 3;

    repeated int32 posted_fields = 4 [packed=true];
    repeated float values = 5 [packed=true];

    optional Statistics statistics = 6;
    optional int32 gridSizeZ = 7;
}
```

While streaming, the client sends frame request messages whenever a simulation time step (frame) is received, causing the server to send the most current time step that has been computed since sending the previous one. This ensures that no more messages are sent than can be transferred, which would lead to buffer overruns. To prevent bandwidth from being wasted due to the latency of requesting a new frame only after the previous one has been received, two frames are requested when a new connection is established, which corresponds to double buffering. A larger number of frames could be pre-requested as well (triple buffering or more) to overcome larger transient bandwidth changes. A full evaluation over varying buffer sizes was not performed within the scope of this paper. Using the double buffering approach as described leads to an improvement in bandwidth exploitation of up to 50% compared to the naïve implementation.

Which fields are streamed to the client is determined by a query. The streaming prototype currently supports two query types:

1. Any number of result fields, e.g., `VelocityX` and `VelocityY`.
2. A query expression combining multiple fields into one.

The former is used when individual results are viewed by the user, and when post-processing is performed on the client for evaluation. The latter is forwarded to our query compiler or an interpreter that was implemented for comparison. A query expression consists of identifiers for the respective available results fields, operators or functions combining them, and parentheses for controlling operator order. The identifiers are specific to the simulation back-end and characteristic of the respective physical domain, e.g., `VelocityX`, `VelocityY` or `Pressure` for fluid simulations, or `DisplacementX`, `StressXX` or `StressXY` for structural mechanics simulations. These identifiers can be used to evaluate combinations of multiple fields such as `(VelocityX^2 + VelocityY^2) / 2 + Pressure`, which corresponds to $\frac{|\vec{v}|^2}{2} + p$, the sum of kinetic and static energy densities of a fluid with density $\rho = 1$.

3.4 Query Compiler

The query compiler prototype consists of an expression parser and an LLVM-based, optimizing back-end. Additionally, an interpreter has been implemented. The compiler is packaged as a shared library, for easy reuse on both client and server.

For many optimizations, especially vectorization, the optimizer must have knowledge if pointers to data:

1. ... may alias or not. Aliasing occurs if the same address in memory is reachable via different pointers. Aliasing prevents vectorization, as it can introduce additional dependencies between loop iterations if a pointer to data that is being read from can alias a pointer to data that is written to.
2. ... are aligned or not. Aligned data is allocated with at an adress that is a multiple of a specific power of two. This information is relevant as many vector instruction sets require loads and stores to be aligned to achieve maximum throughput.
3. ... are captured or not. A captured pointer is stored somewhere and may later on be accessed via a different call. This is mostly relevant to callers of a specific function to know if a piece of data remains accessible.
4. ... point to data that is read, written or both. This information is mostly relevant to callers who may want to reorder function calls.

Such information can be passed to LLVM via the use of function and parameter attributes. To maximize the number of optimization opportunities, the CPU back-end generates LLVM intermediate representation code (LLVM-IR) annotated with the appropriate parameter and function attributes according to the LLVM Performance Tips for Frontend Authors[1] (see Listing 2). Specifically, annotating input pointers with the `readonly` and `nocapture` attributes and the output pointer with `noalias`. However, `nocapture` and `readonly` can be inferred by the compiler and did not affect optimization. In previous LLVM versions, the use of `noalias` was necessary to ensure that vectorizing optimizations are not blocked by alias analysis. In the current LLVM

[1] `http://llvm.org/docs/Frontend/PerformanceTips.html`

top-of-tree as of March 2016 vectorized code is generated independent of the presence of the `noalias` attribute. To do so, LLVM adds runtime aliasing checks and a non-vectorized version of the code. However, this increase in code size and the additional check showed no measurable effect on time measurements in our use case. Additionally, alignment annotations (`align` n) can be used so that aligned moves are emitted instead of unaligned moves. Evaluations in a separate test environment with a result field of 4096^2 values did not result in any change in performance on either an Intel Xeon E5-2650 v2 CPU or an Intel Core i7-3770 CPU. In light of this result and as using alignment in the complete process would have required changes to the simulator's allocation strategy, alignment attributes were not used in the final evaluation.

For the GPU back-end, the LLVM NVPTX target was chosen. Alternatively, NVIDIA's proprietary NVVM-IR or OpenCL's SPIR could have been used, as both are based on LLVM-IR as well. NVVM-IR is used with libnvvm [NVI16], NVIDIA's compiler library. libnvvm supports additional proprietary optimizations, which can lead to improved performance. SPIR can be used with OpenCL to support both AMD and NVIDIA GPUs. However, both NVVM-IR and SPIR are based on older LLVM versions. Therefore, using either would mean using two different versions of LLVM for CPU and GPU code, or not having the full range of CPU optimizations, such as vectorization in the presence of potential aliasing, and instruction sets supported in current versions available. In the future, the addition of SPIR-V [Kes15], the binary intermediate representation introduced with Vulkan and OpenCL 2.1, as an additional target for LLVM [Yax15] will make targeting all platforms that support OpenCL significantly simpler.

LLVM's optimization pipeline consists of a set of passes which take LLVM-IR as input and produce transformed LLVM-IR as output, as well as a number of analysis passes. One such pass is the instruction combining pass, which replaces complex instruction sequences by simpler instructions if possible. Among these are transformations that convert calls of math library functions such as `powf` to calls of faster functions such as `sqrtf` for `powf(`x`, 0.5)` or individual floating point instructions for `powf(`x`, 2)`. However, these functions are identified by name and NVIDIA libdevice math library prefixes all names with `__nv`. To make full use of the instruction combining pass for GPU code as well, we generate code using unprefixed calls and run a subset of optimizations (primarily inlining and instruction combining) before retargeting call instructions to the prefixed versions and linking libdevice. After linking, the full set of optimization passes is run.

Listing 2: LLVM IR generated by the query compiler before optimization for an expression equivalent to a `saxpy`-operation.

```
; Function Attrs: alwaysinline nounwind readnone
define private float @kernel(float, float, float)
    #0 {
entry:
  %3 = fmul float %0, %1
  %4 = fadd float %3, %2
  ret float %4
}

; Function Attrs: nounwind
define void @map(i64, float* noalias nocapture,
    float, float* nocapture readonly, float*
    nocapture readonly) #1 {
entry:
  %5 = icmp ult i64 0, %0
  br i1 %5, label %body, label %exit

body:                          ; preds = %body, %
    entry
  %6 = phi i64 [ 0, %entry ], [ %13, %body ]
  %7 = getelementptr inbounds float, float* %3, i64
       %6
  %8 = load float, float* %7
  %9 = getelementptr inbounds float, float* %4, i64
       %6
  %10 = load float, float* %9
  %11 = call float @kernel(float %2, float %8,
      float %10)
  %12 = getelementptr inbounds float, float* %1,
      i64 %6
  store float %11, float* %12
  %13 = add nuw i64 %6, 1
  %14 = icmp ult i64 %13, %0
  br i1 %14, label %body, label %exit

exit:                          ; preds = %body, %
    entry
  ret void
}

attributes #0 = { alwaysinline nounwind readnone }
attributes #1 = { nounwind }
```

Unlike a general purpose, Turing complete programming language, the simple nature of our query expressions ensures that security is easy to maintain. A general purpose language would require sandboxing to disallow certain operations, and ensure that illegal code does not crash the entire system. Additionally, timeouts would be necessary to prevent infinite loops and/or deadlocks from affecting the server. Expressions with no explicit looping constructs and access only to mathematical functions are inherently secure. The only necessary limit is the length of the expression, as an arbi-

trarily long expression can result in an arbitrarily large amount of work.

4 RESULTS

In this section, we analyze the performance of our streaming protocol and our query compiler.

4.1 Hardware Setup

For the evaluation, the simulation server was set up on a dual Intel Xeon E5-2650v2 server (two octa-core processors running at 2.66 GHz) with two NVIDIA GRID K2 graphics cards (4 GPUs total) and 64 GiB RAM running Ubuntu Linux 13.10. The graphical client was installed on an Intel Core i7-2600 (quad-core processor running at 3.4 GHz) desktop workstation with an NVIDIA Geforce GTX 580 GPU and 16 GiB RAM running Windows 7. For the HTML5 client, tests were additionally performed on a OnePlus One smartphone with a Qualcomm Snapdragon 801 CPU (quad-core processor running at up to 2.5 GHz) and 3 GiB RAM running Cyanogen OS 12.1 (based on Android 5.11). To cover both major mobile platforms, tests were also performed on an Apple iPhone 6S with an Apple A9 CPU (dual-core processor running at up to 1.85 GHz) and 2 GiB RAM running iOS 9.2.

4.2 Network Performance and Bandwidth Limitations

Figures 4 and 5 show the system's performance in terms of data throughput and frames per second when transmitting one, two or three fields with different network bandwidths. In this particular example, these fields were `Pressure`, `VelocityX` and `VelocityY` with a size of 1024^2 floating point values each. Bandwidth limiting was realized on the server side using Linux Traffic Control `tc`. Only outgoing bandwidth is limited, but the messages sent by the client are only tens of bytes in size and should therefore not affect the results.

Increasing the available network bandwidth also increases the client's data throughput as well as the achievable frames per seconds, as more data can be transmitted across the network. At the same time, the server's throughput and frame rate drop slightly, because more time is spent serializing messages instead of calculating new results. This decrease could be compensated by implementing double buffering and performing simulation and serialization asynchronously. However, this would lead to increased memory requirements. In all cases, the server's performance is a natural upper limit for the client that cannot be exceeded. When transmitting more than one field, this limit only becomes relevant for client-server configurations in a LAN setup with more than 1 Gbit/s. For a single field, 500 Mbit/s are sufficient to reach full performance. The fixed bandwidth limit itself is never reached, as the limit is applied at the TCP level and the effective bandwidth only includes floating point data and neither other data nor WebSocket and ProtoBuf encoding overheads.

	Time [ms]		
Number of Fields	**1**	**2**	**3**
Serialization	8.81	18.9	30.0
Native (Desktop)	7.89	14.5	20.2
Chrome (Desktop)	86.5	174	254
Firefox (Desktop)	115	221	380
Chrome (Android)	435	841	1202
Safari (iOS)	233	346	516

Table 1: Serialization and deserialization times for various platforms for a varying number of fields. Even on desktop machines, deserializing a single 1024^2 field in JavaScript takes approximately 0.1 seconds.

4.3 Serialization and Deserialization Costs

Another criterion for good performance and smooth visualization is the time required to serialize the results produced on the server and to deserialize the incoming messages on the client. Table 1 shows the serialization and deserialization costs for one, two and three fields with a size of 1024^2 floating point values per field (as in Section 4.2). Each measurement represents an average over 500 simulation steps. Note, that new frames are only transmitted to the client if the processing of the previous frame is finished. For the client, we also tested different scenarios with desktop and mobile environments. As all fields are concatenated for serialization, the required time increases linearly in all cases. While serialization and deserialization take between 7 and 30 milliseconds when using ProtoBuf in a native C++ application, performance decreases significantly when switching to browser-based applications using JavaScript. Although Chrome 47.0 outperforms Firefox 42.0, deserialization times of 86 to 254 milliseconds on a desktop workstation make it challenging to reach interactive frame rates for more than one field.

On mobile devices, deserialization times of 435 or 233 milliseconds for Chrome 46.0 and Safari 601.1, respectively, make interactive frame rates effectively impossible and raise the need to investigate alternative (de)serialization methods (see Section 6).

4.4 Query Compiler

To analyze the performance of our query compiler, compile times and average evaluation times were measured for three query expressions of varying complexity involving a varying number of results fields:

1. The absolute pressure $|p|$:
   `abs(Pressure)`

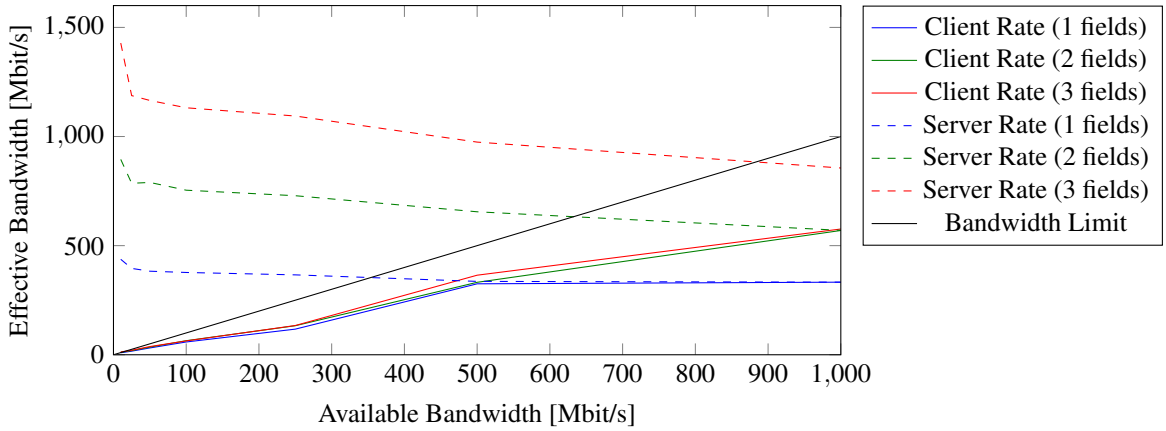


Figure 4: Effective client bandwidths when network bandwidth is limited. The rate at which the server produces data imposes an additional upper limit. This limit decreases with increasing network bandwidth as the server spends more time serializing data and is only reached for bandwidths greater than 500 Mbit/s per field.

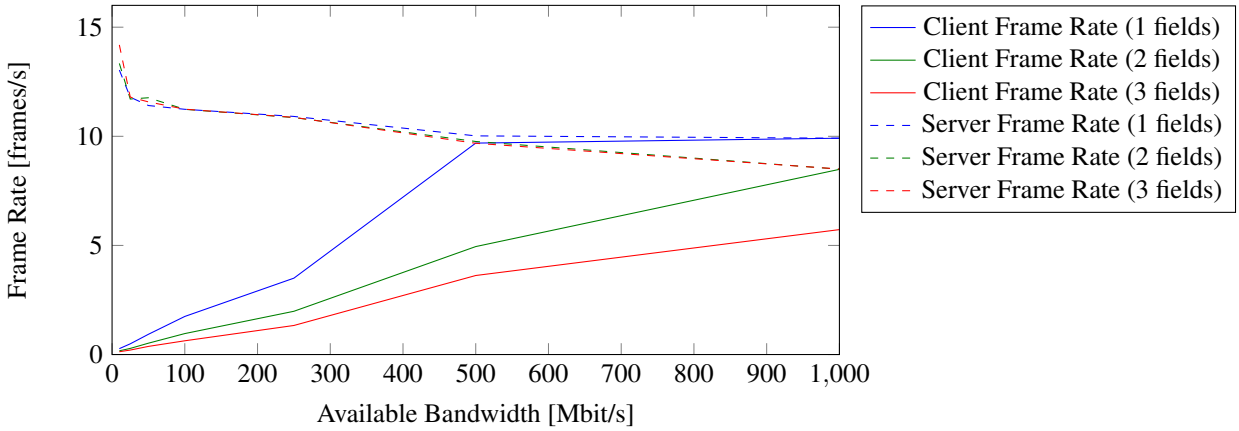


Figure 5: Client and server frame rates for several network configurations. The client frame rate increases with higher network bandwidth, as more data can be sent by the server. As in Fig. 4, the server frame rate limits the client frame rate.

2. The absolute velocity $|\vec{v}|$:
   `sqrt(VelocityX^2+VelocityY^2)`

3. The total energy density $\frac{v^2}{2} + gz + \frac{p}{\rho}$ with $g = 0$ and $\rho = 1$:
   `(VelocityX^2+VelocityY^2)/2 + Pressure`

These expressions were compiled and executed on the server described in Section 4.1. Although this set of example expressions is not exhaustive, it consists of common expressions entered by a user. The absolute value of the pressure can be of interest when the results of a compressible simulation are viewed, as the amplitude of a approximately periodic wave may be of greater interest than its absolute phase. The absolute velocity as the magnitude of a vector field is frequently required and most visualization systems include it as a built-in option. The isocontours of the total energy density are an alternative to streamlines, as according to the Bernoulli equation the total energy density must remain constant along each streamline for incompressible fluids.

All measurements in this section were performed and averaged over 80 runs of simulations on a 1024^2 grid running for 500 frames for each expression. Note that calculation is only performed for frames actually transmitted to the client and that compilation is performed once per simulation run. Therefore, the sample size for the average compilation time is 80 per expression and less than 40000 for the average calculation time.

The measured compile times are shown in Table 2. CPU compilation is completed within less than 10ms and only shows a slight increase depending on expression complexity. Although marginally slower compilation is expected due to the repetition of some optimization passes (see Sec. 3.4), GPU compilation is much slower at over 77 ms and is dominated by a constant component. Further analysis shows that 33% of that time is spent linking libdevice and 61% is spent on the final set of optimization passes. A likely reason for the signif-

	Time [ms]		
Expression	**Interp.**	**CPU**	**GPU**
Expr. 1	0.03	6.60	77.1
Expr. 2	0.04	7.22	77.1
Expr. 3	0.04	9.37	77.2

Table 2: Average compile times for the three example expressions in Sec. 4.4. The times for the interpreter only include expression parsing.

	Time [ms]		
Expression	**Interp.**	**CPU**	**GPU**
Expr. 1	14.1	8.91	4.49
Expr. 2	53.1	13.1	4.27
Expr. 3	63.1	17.1	4.38

Table 3: Average execution times for the three example expressions in Sec. 4.4.

		CPU		GPU	
Expression		**Calc.**	**Copy**	**Calc.**	**Copy**
Expr. 1	**ms**	1.99	1.04	0.10	0.98
	%	65.7	34.3	9.2	90.8
Expr. 2	**ms**	2.20	1.98	0.13	0.97
	%	52.6	47.4	11.6	88.4
Expr. 3	**ms**	1.13	3.02	0.16	0.99
	%	27.2	72.8	13.6	86.4

Table 4: Decomposition of evaluation time into calculation and GPU-CPU transfer times.

		Break-even [Frames]	
Expression		**CPU**	**GPU**
Expr. 1	**Interp.**	2 (1.27)	9 (8.02)
	CPU	—	16 (15.95)
Expr. 2	**Interp.**	1 (0.18)	2 (1.58)
	CPU	—	8 (7.91)
Expr. 3	**Interp.**	1 (0.20)	2 (1.31)
	CPU	—	6 (5.33)

Table 5: Break-even points of using CPU or GPU JIT compilation instead of an interpreter and GPU instead of CPU JIT compilation for the three example expressions in Sec. 4.4. The numbers are computed from the measurements in Tables 2 and 3 and rounded up to the nearest integer. Break-even before rounding is shown in parentheses.

icant increase in optimization time is the much larger module due to the size of libdevice.

The measured calculation times are shown in Table 3. It can be seen that the timings for the GPU version are approximately constant for all three expression, whereas for the CPU version they grow with the number of fields used in the expression. To determine the reasons for this behavior, additional measurements decomposing the total evaluation time into computation and data transfer times were performed. Table 4 shows the results of these measurements that were performed on a desktop machine equipped with an Intel Core i7-3770 CPU with 3.40 GHz and an NVIDIA GeForce GTX 580 GPU. In the case of CPU evaluation, all relevant fields have to be copied from the GPU depending on the expression used. This is reflected in the linear increase in copy times and explains the dependency seen in Table 3. For GPU evaluation, only the derived field has to be copied to system RAM. As the GPU evaluation times are dominated by the expression-independent copy component, the total evaluation time is approximately constant, as seen in Table 3 and leads to an improvement of up to 72.3% for Expr. 3.

The total time to process n simulation frames (time steps) is $t_c + nt_e$, where t_c is the compilation time, t_e is the average execution time per frame and n is the number of frames executed. Therefore the break-even between two methods a and b can be computed as $n = \left\lceil \frac{t_{c,a} - t_{c,b}}{t_{e,b} - te,a} \right\rceil$. Table 5 summarizes the different break-even points of using just-in-time (JIT) compilation instead of an interpreter. In all but the first case, the cost of compilation for CPU is amortized within the first frame, as the sum of compilation and execution time for one frame is less than the execution time for the interpreter. Due to the large compilation overhead, the break-even point of using the GPU instead of the CPU occurs significantly later. The break-even point of using the GPU instead of the CPU is reached after less than 10 frames for Expressions 2 and 3. As compilation time is independent of field size and execution time depends linearly on it, the break-even point will be reached even more quickly for larger simulation domains.

5 CONCLUSION

Using the query compiler introduced in Section 3.4, only one result field has to be sent to the client. This ensures that a high visualization frame rate can be achieved with bandwidths as low as 500 Mbit/s (see Sec. 4.2), allowing the user to view more current data. Additional latency and computation costs due to deserialization are avoided as well, making HTML5 clients feasible on desktop workstations (see Sec. 4.3). By using an optimizing compiler, server CPU and GPU times are reduced by a factor of up to 14 compared to the naïve approach of using an interpreter (see Sec. 4.4). As computation time and required bandwidth directly affect visualization latency, research question 1 *"Can compiler technologies be used to decrease visualization latencies in a remote scientific visualization system?"* can be answered positively.

The second research question *"Can GPU-based simulations running at interactive rates profit from GPU-based query compilation?"* can be confirmed as well. By computing derived expressions directly on the GPU, a significant amount of time can be saved. By only copying a single field independent of the number of fields used in the expression, the amount of data transferred over the PCIe bus can be reduced, as shown in

Sec. 4.4. Additionally, computation speed is increased by a factor of up to 20 compared to the CPU.

In summary, we have performed a detailed analysis of streaming performance and shown that optimizing compiler technologies such as LLVM can be used to significantly improve performance and reduce bandwidth costs for streaming visualization of interactive simulations. By additionally moving data transformation work to the GPU, the costs of PCIe bus transfers can be minimized as well for GPU-based simulation back-ends.

Compared to MapD (see Sec. 2.1) we have taken a similar approach of leveraging compiler technologies for visualization, but applied it to interactive scientific visualization instead of visual analytics. The range of available options is currently significantly smaller, but further enhancements are outlined in the following section.

Compared to application sharing (see Sec. 2.3) our approach of pre-transforming simulation data on the server before transmitting it to the client for final visualization has both benefits and drawbacks. Many interactions relevant during exploration of simulation results, including color map changes and panning/zooming in 2D or camera position in 3D, can now be performed without any network round trip latency using our approach. Application sharing always incurs at least one network round trip for all user interactions. However, the time to first image is potentially higher, as floating point simulation data is frequently larger than the resulting image compressed using a video codec. This also decreases the number of frames per second that can be transmitted given a limited bandwidth. This drawback can be offset by applying compression methods as well (see Sec. 2.2). Furthermore, the portion of simulation data that is transmitted could be limited to the visible part and resolution, however this limits panning/zooming or can create temporary holes in the visualization that are fixed as soon as an updated frame is received.

6 FUTURE WORK

Several potential extensions could be implemented to improve performance further or increase flexibility. Compression algorithms including those presented in [OB11] or [Lin14] can be added to further reduce bandwidth requirements at the cost of additional processing on both client and server. Queries could be extended to support subfields, i.e., named boundaries or subdomains, for instance an inlet in a CFD simulation or a specific component in a CSM simulation. Especially in combination with reductions, for example averages or maximums of fields, such subfield queries could become useful. However, parallel reductions as required by the GPU back-end require reimplementation of many scalar optimizations such as common subexpression elimination, as parallelism can not be expressed directly in LLVM-IR. Expressing parallelism in LLVM is a topic of ongoing research (see, e.g., [Kha+15]). Furthermore, calculations involving matrices and tensors would be useful for several physical domains, including CSM. Fields could also be annotated with physical units to detect mistakes due to adding fields with mismatched units.

Considering the bad JavaScript performance, alternative serialization formats promising lower deserialization costs such as Cap'n Proto [San16] or FlatBuffers [Goo16], or JavaScript's native JSON (JavaScript object notation) format could be investigated. However, these typically come at an increased bandwidth cost.

ACKNOWLEDGEMENTS

This work has been supported in part by the EU project CloudFlow (FP7-2013-NMP-ICT-FoF-609100).

REFERENCES

[Alt+16] Christian Altenhofen et al. "Rixels: Towards Secure Interactive 3D Graphics in Engineering Clouds". In: *Transactions on Internet Research (TIR)* 12.1 (Jan. 2016), pp. 31–38. ISSN: 1820-4503.

[Che+11] Kuan-Ta Chen et al. "Measuring the Latency of Cloud Gaming Systems". In: *Proceedings of the 19th ACM International Conference on Multimedia*. MM '11. Scottsdale, Arizona, USA: ACM, 2011, pp. 1269–1272. ISBN: 978-1-4503-0616-4. DOI: 10.1145/2072298.2071991.

[Chi+12] Charisee Chiw et al. "Diderot: A Parallel DSL for Image Analysis and Visualization". In: *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI '12. Beijing, China: ACM, 2012, pp. 111–120. ISBN: 978-1-4503-1205-9. DOI: 10.1145/2254064.2254079.

[Cho+14] Hyungsuk Choi et al. "Vivaldi: A Domain-Specific Language for Volume Processing and Visualization on Distributed Heterogeneous Systems". In: *Visualization and Computer Graphics, IEEE Transactions on* 20.12 (Dec. 2014), pp. 2407–2416. ISSN: 1077-2626. DOI: 10.1109/TVCG.2014.2346322.

[Duk+09] D.J. Duke et al. "Huge Data But Small Programs: Visualization Design via Multiple Embedded DSLs". English. In: *Practical Aspects of Declarative Languages*. Ed. by Andy Gill and Terrance Swift. Vol. 5418. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 31–45. ISBN: 978-3-540-92994-9. DOI: 10.1007/978-3-540-92995-6_3.

[Goo08] Google. *Protocol Buffers (protobuf)*. Website, retrieved 2016-03-14. 2008. URL: https://github.com/google/protobuf.

[Goo16] Google. *Flatbuffers*. Website, retrieved 2016-03-14. 2016. URL: https://github.com/google/flatbuffers.

[Gri13] Ilya Grigorik. *High Performance Browser Networking*. O'Reilly Media, 2013. ISBN: 978-1-4493-4476-4.

[Kes15] John Kessenich, ed. *SPIR-V Specification*. Version 1.00, Rev. 2. Khronos Group, Nov. 2015. URL: https://www.khronos.org/registry/spir-v/specs/1.0/SPIRV.pdf.

[Kha+15] Dounia Khaldi et al. "LLVM Parallel Intermediate Representation: Design and Evaluation using OpenSHMEM Communications". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM '15. ACM, Nov. 2015, 2:1–2:8. DOI: 10.1145/2833157.2833158.

[LA04] Chris Lattner and Vikram Adve. "LLVM: A Compilation Framework for Lifelong Program Analysis and Transformation". In: *Code Generation and Optimization, 2004. CGO 2004. International Symposium on*. Mar. 2004, pp. 75–88. DOI: 10.1109/CGO.2004.1281665.

[Lin14] P. Lindstrom. "Fixed-Rate Compressed Floating-Point Arrays". In: *Visualization and Computer Graphics, IEEE Transactions on* 20.12 (Dec. 2014), pp. 2674–2683. ISSN: 1077-2626. DOI: 10.1109/TVCG.2014.2346458.

[Map16] MapD Technologies, Inc. *MapD*. Website, retrieved 2016-03-14. 2016. URL: http://www.mapd.com/.

[Mic16] Microsoft. *Remote Desktop Protocol*. Website, retrieved 2016-03-14. 2016. URL: https://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx.

[MŞ15] Todd Mostak and Alex Şuhan. *MapD: Massive Throughput Database Queries with LLVM on GPUs*. Website, retrieved 2016-03-14. June 2015. URL: https://devblogs.nvidia.com/parallelforall/mapd-massive-throughput-database-queries-llvm-gpus.

[Nic+08] John Nickolls et al. "Scalable Parallel Programming with CUDA". In: *ACM Queue* 6.2 (Mar. 2008), pp. 40–53. ISSN: 1542-7730. DOI: 10.1145/1365490.1365500.

[NVI16] NVIDIA. *CUDA LLVM Compiler*. Website, retrieved 2016-03-14. 2016. URL: https://developer.nvidia.com/cuda-llvm-compiler.

[OB11] Molly A. O'Neil and Martin Burtscher. "Floating-point Data Compression at 75 Gb/s on a GPU". In: *Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units*. GPGPU-4. Newport Beach, California, USA: ACM, 2011, 7:1–7:7. ISBN: 978-1-4503-0569-3. DOI: 10.1145/1964179.1964189.

[Pey03] Simon Peyton Jones, ed. *Haskell 98 Language and Libraries: The Revised Report*. Cambridge University Press, 2003. ISBN: 978-0-521-82614-3.

[Rau+14] P. Rautek et al. "ViSlang: A System for Interpreted Domain-Specific Languages for Scientific Visualization". In: *Visualization and Computer Graphics, IEEE Transactions on* 20.12 (Dec. 2014), pp. 2388–2396. ISSN: 1077-2626. DOI: 10.1109/TVCG.2014.2346318.

[Ric+98] Tristan Richardson et al. "Virtual network computing". In: *IEEE Internet Computing* 2.1 (Jan. 1998), pp. 33–38. DOI: 10.1109/4236.656066.

[RKB06] Paruj Ratanaworabhan, Jian Ke, and Martin Burtscher. "Fast lossless compression of scientific floating-point data". In: *Proceedings of the Data Compression Conference*. DCC '06. 2006, pp. 133–142. DOI: 10.1109/DCC.2006.35.

[San16] Sandstorm.io. *Cap'n Proto*. Website, retrieved 2016-03-14. 2016. URL: https://capnproto.org/.

[TAS06] N. Tolia, D.G. Andersen, and M. Satyanarayanan. "Quantifying interactive user experience on thin clients". In: *Computer* 39.3 (Mar. 2006), pp. 46–52. ISSN: 0018-9162. DOI: 10.1109/MC.2006.101.

[Web+15] Daniel Weber et al. "A Cut-Cell Geometric Multigrid Poisson Solver for Fluid Simulation". In: *Computer Graphics Forum* 34.2 (May 2015), pp. 481–491. ISSN: 0167-7055. DOI: 10.1111/cgf.12577.

[Yax15] Liu Yaxun. *[RFC] Proposal for Adding SPIRV Target*. Website, retrieved 2016-03-14. June 2015. URL: http://lists.llvm.org/pipermail/llvm-dev/2015-June/086848.html.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

3D segmentation of the tracheobronchial tree using multiscale morphology enhancement filter

Samah Bouzidi
Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence
ReGIM,University of Sfax, Tunisia
sbouzidi@labri.fr

Fabien Baldacci
Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence
fabien.baldacci@labri.fr

Chokri Ben Amar
Research Group in Intelligent Machines(ReGIM), University of Sfax , Tunisia
chokri.benamar@ieee.org

Pascal Desbarats
Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence
pascal.desbarats@labri.fr

ABSTRACT

In this article we present a new region growing algorithm for airway segmentation based on multiscale black top-hat enhancement filter. Lung airways are tubular structures that display specific characteristics, such as highly variable intensity levels within the lumen and proximity to vessels. The proposed airways enhancement filter aims to separate airways from adjacent lung parenchyma and vessel. Based on the filter ouput, the region growing is performed in order to delineate the airways and then to reconstruct the tracheobronchial tree. The proposed method has been applied on various CT scans. In this paper, an experimental comparison study between our filter and the "gold standard" filters used to enhance tubular structures (Frangi, Sato and Krissian filters) followed by a region growing process is performed on data from the *VESSEL12* challenge framework. Our approach outperforms the other considered methods in terms of retrieved bronchi and computing time.

Keywords

bronchial segmentation, enhancement filter, lung, CT chest scan.

1 INTRODUCTION

Visualisation and analysis of human tracheobronchial tree (TBT) is crucial for many clinical procedures. The assessments of the airways tree structure and the monitoring of lung interventions requires a good knowledge of the airway morphometry such as airway wall thickness and lumen diameter. With the introduction of Computed Tomography (CT) scanning into the non-invasively assessing of lung abnormalities, a 3D extraction and visualisation of the TBT has become more important. Anatomically, the tracheobronchial tree consists of a network of hollow branching tubes that enable airflow to go into the lungs through the main airways, the trachea. Boyden [3] proposed to decompose the TBT to 23 generations where generation zero corresponds to the trachea, the 3^{rd} generation to the segmental bronchus, the 4^{th} to the subsegmental bronchus and the 23^{rd} generation corresponds to terminal bronchioles. As the tree penetrates deeper into the lungs, the airways size decrease, e.g the 7^{th} branching generation can have diameters in the *mm* range. This complex branching structure makes its manual extraction process tedious, time consuming and changing across image analysts. A large amount of airways analysis has been reported in the literature. Most of proposed methods of airways tree segmentation are based on the extraction of the airway lumen which appears in CT chest scan as a dark tube surrounded by a bright airway wall. Based on that assumption, numerous approachs [4, 12, 17, 18] used region growing process to segment the tree. Starting from a seedpoint within the trachea, voxels are added to the process if its X-ray density belong to the lumen density range. However, intensity based region growing deals with many difficulties and often leads to leakage into the lung parenchyma. First, there is no standard lumen intensity range since CT scans may be acquired under different scanning conditions and/or depict different diseases. Second, as noise and partial volume effects decrease the contrast between the air and the surrounding tissue then the whole lung can be added to the growing region. Last but not least, growth can also be interrupted earlier in case of lung disease (e.g., emphysema). The segmentation process is blocked in distal bronchial generation. Despite

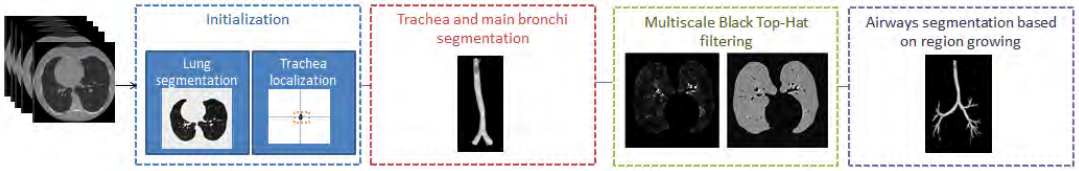


Figure 1: Overview of the proposed method.

of these limitations and due to its simple implementation, region growing is still the most popular approach to segment airways tree. To overcome their problems, several strategies have been proposed to improve region growing results.

In this paper, region growing are performed twice. First, an intensity based region growing is employed to segment trachea and main branchi. Then, the input volume is enhanced using the multiscale Black Top-Hat filter. Therefore, small airways wall becomes more distinguish from background. Thereafter, the second region growing is performed on the processed volume to extract the TBT and prevent leakage. The content of this paper may be summarized as follows. In section 2, an overview of existing airways segmentation approachs are presented. In section 3, the proposed method is explained in detail. Section 4 presents the experimental results. Finally, conclusions and perspectives are drawn in section 5.

2 RELATED WORKS

A lots of efforts have been made to prevent region growing from leaking into the lung parenchyma by adjusting the growing criterion. Mori et al. [13] proposed explosion-controlled region growing algorithm that updates iteratively the intensity threshold until parenchymal leakage (explosion) is detected. Fabijanska [4] used two passes of 3D seeded region growing where the second one is guided by a morphological gradient information that allows to locally identify airways region and prevents the process from leakage. Weinheimer et al. [19] employed an adaptive region growing approach constrained by airways lumen and wall intensities thresholds and performed in axial, coronal, and sagittal plane. Similarly to Mori's method [13], Lee et al. [11] proposed an adaptive region growing method applied within localized cylindrical volumes in order to control the segmentation process. Other works address the RG leakage problem by filtering the image before performing the tree segmentation. In that approach, tubular enhancement filters based on hessian matrix analysis [12, 17] and mathematical morphology operations [1, 14, 9] are used to isolate candidate airway locations. In the work of Lo et al. [12] the growing criterion is based on an airways classifier and vessel orientation similarity that use hessian matrix analysis. First, Hessian eigenvalues analysis are employed twice to differentiate airways and vessel voxels. From obtained vessel voxel, Hessian eigenvector analysis is performed to define neighboring airways orientation. Aykac et al [1] and Pisupati et al. [14] used grayscale mathematical morphology to identify candidate airways on 2-D CT slices. The grayscale reconstruction is performed using different sized structure elements (SE) in order to detect airways over a wide range of sizes. Airways tree is then reconstructed using slice by slice region growing. Similarly, Irving et al. [9] applied multiscale morphological filtering in the axial, sagittal and coronal planes of the volume. After thresholding the enhanced volume, airways are segmented using 3D bounded space dilation region growing.

3 MATERIAL AND METHODS

The proposed algorithm consists of four steps and the whole process can be seen in the Figure 1, The input is a 3-D X-ray CT image volume that displays all the structures in the patient's chest, including lungs. The algorithm starts by extracting lungs. The obtained volume is firstly used to perform the region growing and segment main bronchii and is secondly improved by the multiscale Black Top-Hat filter in order to perform the second region growing that adds small bronchi.

3.1 Lung segmentation

Lungs segmentation is performed using a simplified version of Hu et al. [8] and Heuberger et al. [7] algorithms. First of all, the input image is thresholded to separate low-intensity pixels (lung and surrounding air voxels) from high-intensity voxels (hard and soft tissues voxels), the obtained image is illustrated in Figure 2.(b). Then the surrounding air, which is the set of pixels connected to image borders, is identified and removed from the lung volume (see Figure 2.(c)). After that, the lung mask is created by cleaning the interior of lung from noise and airways using morphological closing operation. Finally, the obtained mask is applied on the initial volume to extract lungs and trachea.

3.2 Main bronchi segmentation

3.2.1 Trachea localization

The trachea is localised using the output of the lung segmentation algorithm. The slices are orientated using the DICOM header information (header first/feet first flag) and we search the first slice that contains voxels assigned to the lung region. An horizontal pass through

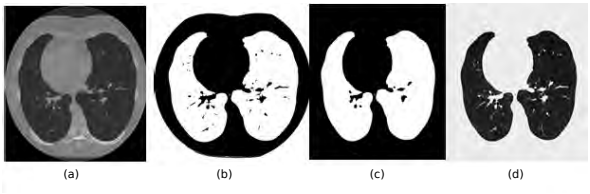


Figure 2: Lung segmentation steps: (a) original, (b) thresholding, (c) background removal and mask creation, (d) lung extraction.

the extracted slice is performed in order to extract trachea voxels as shown in Figure 3. The center of the trachea is the pixel located in the middle of the trachea voxels set. We use this voxel as the seed point for the following region growing process.

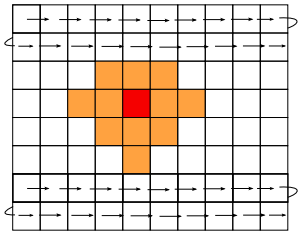


Figure 3: Trachea localization. Trachea pixels set is marked in yellow and trachea seed point is marked in red.

3.2.2 *First region growing process*

After the seed voxel of the region growing algorithm is determined, all their 26-connected neighbours are added to the growing process in order to initialize the growing criterion. We define this criterion as the range between the minimum intensity value of a CT image and the maximum intensity value of added voxels. After that, the 3D intensity based region growing is performed and the upper bound of the growing criterion is adjusted as the initialization step.

The algorithm is stopped when the number of bifurcation is equal to two, which mean that trachea and one of the main bronchi are extracted. The second one will be fully extracted by the second region growing process.

3.3 Multiscale Black Top-Hat filtering

As stated in section 2, graylevel morphological techniques have been widely used to enhance the airways in CT slices. In our case, we use a Black Top-Hat transform (BTH) [6] embedded in a multiscale framework to identify the airways location. In other words, the proposed multiscale Black Top-Hat algorithm integrates the idea of iteratively increasing the structuring element (SE) size to capture smallest and largest bronchi in the lung.
We define the multiscale structuring elements set $\{B_1, B_2, B_3, ..., B_n\}$ as a set of binary diamond SE with increasing size where B_i is the result of i^{th} dilation as follows:

$$B^i = B \oplus B \oplus ... \oplus B \tag{1}$$

Airways extracted at the i^{th} scale by the BTH can be expressed as follows:

$$A_i = I \bullet B^i - I \tag{2}$$

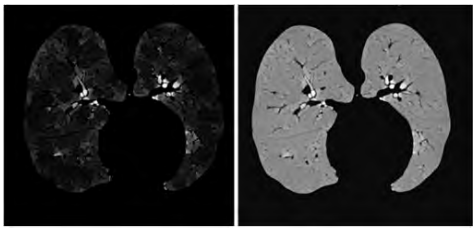


Figure 4: Airways highlighted using the Black Top-Hat transform. Left: the multiscale response of the filter applied in axial slice. Right: its corresponding image difference.

For each slice, the corresponding BTH enhanced is obtained after combining the airways location extracted at each scale (see eq 3). Then, the union of this series of images is taken to form the final Enhanced Airways volume (EA) as described in eq 4.

$$A = \bigcup_i A_i \tag{3}$$

$$EA = \bigcup_z A \tag{4}$$

A grayscale difference volume D is then computed to distinguish more airway locations from lung parenchyma and vessels. Figure 4 illustrates the application of the BTH on the input image and its corresponding image difference in the volume D.

3.4 Second region growing

The second 3D region growing aims to add lung bronchi to the volume defined in 3.2.2. The growing process is performed on the enhanced BTH volume obtained from the previous step. We define the seeds points as the set of points obtained after the first segmentation. Voxels are added to the final volume if their intensities and the intensities of all their neighbours belong to the rangs of enhanced airway lumen. The range was chosen experimentally.

4 RESULTS

In this section, our method's efficiency is evaluated by comparison to a rough region growing with manually selected threshold results and to state-of-the-art

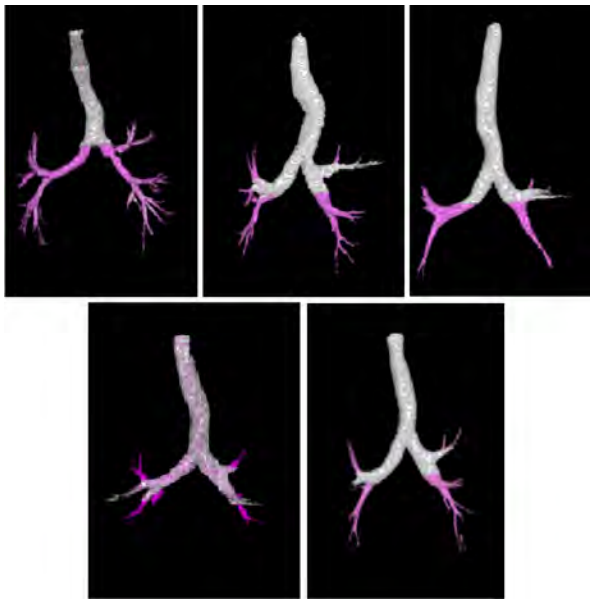


Figure 5: Segmentation results of airway tree segmentation using proposed method and the raw region growing. Airways marked in white are those extracted by the raw region growing. Airways segmented during the proposed algorithm are assigned with pink colour.

Hessian-based vessel enhancement filters. The first filter is the "gold-standard" Frangi vesselness filter [5]. The second is the Sato's line filter [16] and the third is the medialness Hessian-based vesselness filter derived from the work of Krissian et al. [10]. All filters are presented in section 4.2.2. All computations were performed on an intel-Xeon E3-1200 @ 3.60GHz, 16GB RAM, Ubuntu Linux 64 bit.

4.1 Clinical data

We have first assessed qualitatively our method using various CT chest scan. Then, we have used data from the *VESSEL12* challenge (http://vessel12.grand-challenge.org/) for the quantitative analysis of airway tree segments of each method. The evaluation database included five pairs of anonymized MSCT cases acquired using several CT scanners and protocols [15]. Table. 1 presents the characteristics of each scan.

4.2 Segmentation results evaluation

Results of applying proposed algorithm to the first five *VESSEL12* CT data sets are illustrated in Figure 5. In the context of airways segmentation, the assessment of segmentation results is a tedious task if the gold standard isn't provided. A manual segmentation can be a good alternative except however the operation is very time consuming and requires expert's skills. In our case, as the gold standard isn't available, we compare our results with the TBT trees obtained when the data is enhanced or not.

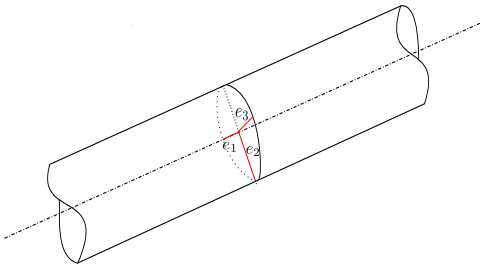


Figure 6: The eigenvalues e_2 and e_3 of the Hessian matrix define the principal curvature of the tube [2].

4.2.1 Proposed method vs raw RG

We first compare the performance of our algorithm to the raw region growing (RRG) algorithm. Intensities range parameters of the later are selected for each scan manually in order to avoid leakage. As illustrated in Figure 5.(a) our algorithm successfully extend from the first generation which is the final generation obtained by the RRG algorithm to the sixth generation. The added bronchi are presented in pink while the branches detected by both algorithms are shown in white.

4.2.2 Proposed method vs Hessian based enhancement filters

We first present in what follow the theory behind the three enhancement filters used for comparison propose.

Frangi line filter. Frangi et al. [5] perform a Hessian eigenvalue analysis to enhance voxel within tubular structures (vessels, airways...). Based on the information that dark tubular structures have two positive larger eigenvalues ($e_3 > 0$ *and* $e_2 > 0$) and the third eigenvalue being close to zero ($e_1 \approx 0$). The proposed line filter is defined as:

$$T(x) = \begin{cases} ((1-exp(\frac{R_A^2}{2\alpha^2}))exp(\frac{R_B^2}{2\beta^2})(1-exp(\frac{S^2}{2\gamma^2})) \\ 0, e_3 < 0 \; and \; e_2 < 0 \end{cases} \tag{5}$$

with $R_A = |\frac{e_2}{e_3}|$, $R_B = \frac{|e_1|}{\sqrt{e_2 e_3}}$ and S is the Frobenius norm of the Hessian matrix. α, β and γ control the sensitivity of the filter to R_A , R_B and S measures.

Sato line filter. Similar to the work of Frangi et al. [5]. Sato et al. [16] proposed the following line filters to enhance tubular structures:

$$T(x) = \begin{cases} exp(-\frac{e_1^2}{2(\alpha_1 e_c)^2}) & e_1 \leq 0 \; and \; e_c \neq 0 \\ exp(-\frac{e_1^2}{2(\alpha_2 e_c)^2}) & e_1 > 0 \; and \; e_c \neq 0 \\ 0 & e_c = 0 \end{cases} \cdot \tag{6}$$

with $e_c = min(e_2, e_3)$, and α_1 and α_2 are control parameters.

Scan	Image type	Spacing(mm)	Z-spacing (mm)	number of slices	kV/mAs
01	Angio-CT	0.76	1	355	120/40
02	Chest CT	0.71	0.7	415	140/74
03	Chest CT	0.62	0.7	534	120/77
04	LD Chest CT	0.86	1	426	100/44a
05	Chest CT	0.72	0.7	424	140/73

Table 1: Description of the first five CT scans of the *VESSEL12* challenge dataset. Angio-CT: CT with contrast agent, LD: Low Dose [15].

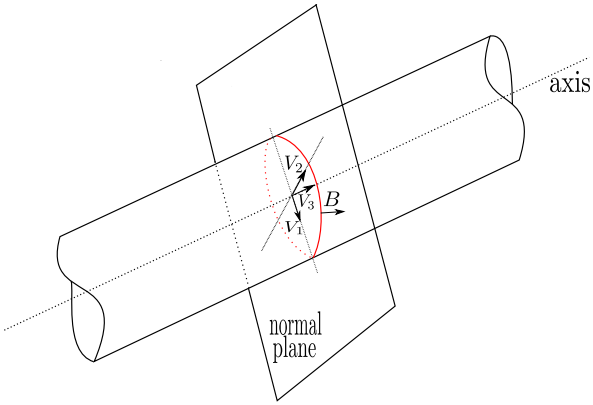


Figure 7: The medialness response obtained from the boundary information (red circle). The Cross-section plane of the tube is spanned by the eigenvectors v_1 and v_2 of the Hessian matrix [2].

Krissian medialness filter. Krissian et al. [10] proposed a medialness function which measures the degree to belong to the medial axis. The response function is estimated by measuring the boundary information at a circular neighborhood which radius is the used scale. The proposed medialness function is represented as follows:

$$R(X,\sigma,\theta) = \frac{1}{N}\sum_{i=0}^{N-1} \mid \bigtriangledown I^{\sigma}(X+\theta\sigma v_{\alpha_i}) \mid \qquad (7)$$

Here, $X = (x,y,z)^T$ is a pixel point, $I^{\sigma}(X)$ is the image at the scale σ, N is the number of samples. The circle is defined by eigen vectors v_1 and v_2 and the radius $r = \sigma\theta$.

Segmentation results. We have used the pipeline of our algorithm to extract trees from the filtred data of each filter. We denote T_{RGF}, T_{RGS}, T_{RGK} respectively the tree obtained with region growing based on Frangi, Sato and krissian filter. We denote the TBT trees obtained by our method T_{RGTH}. Figure 10 summarize the detected airway trees (T_{RGF}, T_{RGS}, T_{RGK} and T_{RGTH}) for each scan sorted by generation number (bronchi order). First order division corresponds to the trachea.

We illustrate also in Figure 8 the obtained tree for each algorithm. From Figure 10 and Figure 8 we can clearly notice that, for all of tested subjects, results obtained by our algorithm were significantly better than those

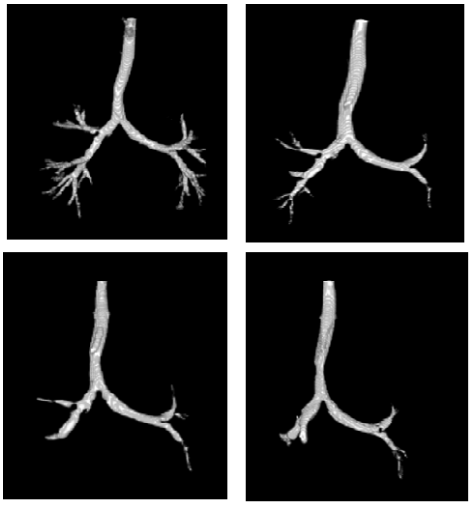


Figure 8: From left to right segmentation results of each algorithm: T_{RGTH},T_{RGF}, T_{RGS}, T_{RGK}.

obtained by other algorithms. Airway trees obtained using proposed segmentation algorithm were more expanded and contained more branches.
Moreover, the proposed method is robust in the sense that it yields good results on different types of scans (low-dose, CT with contrast agent and regular dose). Low-dose CT scans are increasingly utilized to quantify lung disease. In the fourth CT scan which is a Low-dose scan, our algorithm outperforms Frangi and Sato based region growing in terms of retrieved bronchi and Krissian based region growing in terms of generation number as well as detected bronchi.

In terms of runtime, Table 2 depicts the runtime in seconds of each algorithm performed on the first data set. We have used the same number of scales for all filters. Standard region growing algorithm is the fastest because it extracts at most 3 generations (30s). Our algorithm is ranked second with 840s and it is 2 minutes faster than the third one, the Frangi based region growing. The slowest algorithm is the Krissian based region growing.

4.3 Work in progress

In our current work, we are looking for increasing the number of detected generations and improving the rate of bronchi recognized per generation.

Tree	Time (in second)
T_{RGTH}	**840**
T_{RGF}	985
T_{RGS}	1020
T_{RGK}	1140

Table 2: Algorithms runtime.

Generation	Added bronchi
5*th*	4
6*th*	19
7*th*	12
8*th*	4

Table 3: Added bronchi per generation.

For this reason, we have analysed the Black Top-Hat response for each scale. We found that the region growing criterion excluded several bronchi improved by the filter. Excluded bronchi are those recognized by the three smaller scales but don't fill in the criterion range of the region growing. We calculated the BTH using the first three structural elements. The filter response for each SE is then thresholded and combined in a single volume. The obtained volume is added to the volume (see section 3.3) thresholded using the threshold employed in the second pass region growing. As illustrated in Figure 9, the segmentation is improved in terms of generation and in terms of number of retrieved bronchi. Table 3 presents the added bronchi at each generation in the second scan.

5 CONCLUSION AND PERSPECTIVES

In this article, we have presented a new approach based on 3D region growing to segment the bronchial tree. The algorithm is performed on the output of a multi-scale Black Top-Hat filter. It allows to highlight large as well as small bronchi while main bronchi and trachea are first extracted using a standard region growing. The proposed filter guides and constraints the growing process to identify airways region without leaking to the parenchyma region. The algorithm was tested using on different CT scan and it was compared to other region growing based methods using *vessel12* challenge data.

Experimental results show that our methods yields better results than those obtained by the four other methods in terms of the number of retrieved generation and runtime. Even if the method failed to extract bronchi after the seventh generation, our RG didn't leak into parenchyma and extract the TBT in few minutes. Therefore, it seems to be possible to complete the growing process with an advanced local tracking method on each bronchi which will be able to increase tree's depth. Future works will focus on the implementation of a complete segmentation pipeline in which the proposed method will be used as an initialization of the following extraction process.

Figure 9: Modified segmentation result in the first and second scan, added airways are marked in yellow.

6 REFERENCES

[1] D. Aykac, E. A. Hoffman, G. McLennan, and J. M. Reinhardt. Segmentation and analysis of the human airway tree from three-dimensional x-ray ct images. *Medical Imaging, IEEE Transactions on*, 22(8):940–950, 2003.

[2] C. Bauer and H. Simpson. Segmentation of 3d tubular tree structures in medical images. 2010.

[3] E. A. Boyden. *Segmental anatomy of the lungs: a study of the patterns of the segmental bronchi and related pulmonary vessels*. Blakiston Division, McGraw-Hill, 1955.

[4] A. Fabijańska. Two-pass region growing algorithm for segmenting airway tree from mdct

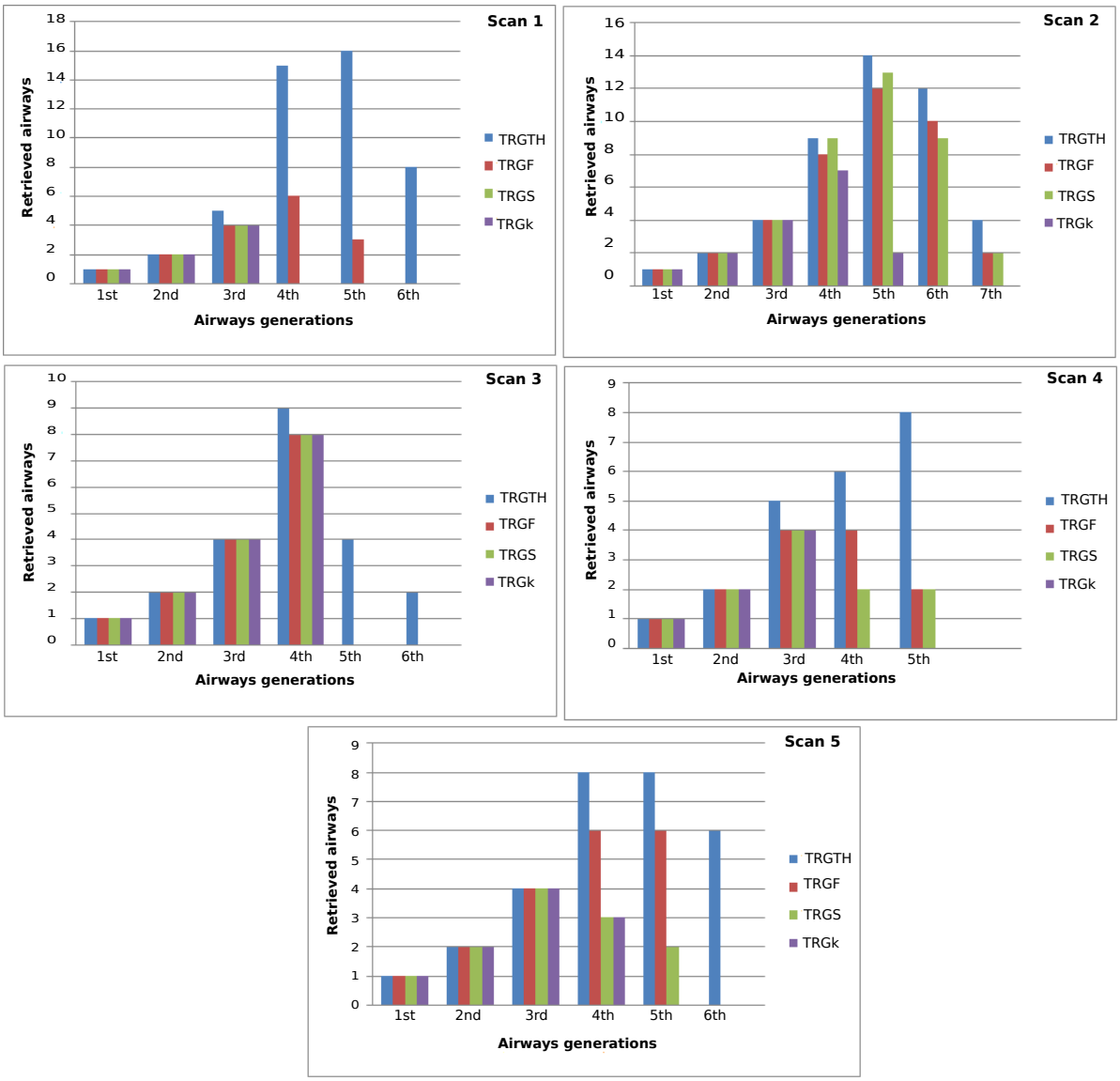


Figure 10: Number of detected airways sorted by generation number. The scan number is indicated on each histogram (cf. Table 1).

chest scans. *Computerized Medical Imaging and Graphics*, 33(7):537–546, 2009.

[5] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention MICCAI98*, pages 130–137. Springer, 1998.

[6] R. C. Gonzalez et al. Re woods, digital image processing. *Addison–Wesely Publishing Company*, 1992.

[7] J. Heuberger, A. Geissbühler, and H. Müller. Lung ct segmentation for image retrieval. *Medical Imaging and Telemedicine*, 2005.

[8] S. Hu, E. A. Hoffman, and J. M. Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images. *Medical Imaging, IEEE Transactions on*, 20(6):490–498, 2001.

[9] B. Irving, P. Taylor, and A. Todd-Pokropek. 3d segmentation of the airway tree using a morphology based method. In *Proceedings of 2nd international workshop on pulmonary image analysis*, pages 297–07, 2009.

[10] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-based detection of tubular structures in 3d images. *Computer vision and image understanding*, 80(2):130–171, 2000.

[11] J. Lee and A. P. Reeves. Segmentation of the airway tree from chest ct using local volume of interest. In *Proc. of Second International Workshop on Pulmonary Image Analysis*, pages 273–284, 2009.

[12] P. Lo, B. Van Ginneken, J. M. Reinhardt,

T. Yavarna, P. A. De Jong, B. Irving, C. Fetita, M. Ortner, R. Pinho, J. Sijbers, et al. Extraction of airways from ct (exact'09). *Medical Imaging, IEEE Transactions on*, 31(11):2093–2107, 2012.

[13] K. Mori, J.-i. Hasegawa, J.-i. Toriwaki, H. Anno, and K. Katada. Recognition of bronchus in three-dimensional x-ray ct images with applications to virtualized bronchoscopy system. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 528–532. IEEE, 1996.

[14] C. Pisupati, L. Wolff, E. Zerhouni, and W. Mitzner. Segmentation of 3d pulmonary trees using mathematical morphology. In *Mathematical morphology and its applications to image and signal processing*, pages 409–416. Springer, 1996.

[15] R. D. Rudyanto, S. Kerkstra, E. M. Van Rikxoort, C. Fetita, P.-Y. Brillet, C. Lefevre, W. Xue, X. Zhu, J. Liang, İ. Öksüz, et al. Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the vessel12 study. *Medical image analysis*, 18(7):1217–1232, 2014.

[16] Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig, S. Yoshida, and R. Kikinis. 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In *CVRMed-MRCAS'97*, pages 213–222. Springer, 1997.

[17] M. Sonka, W. Park, and E. A. Hoffman. Rule-based detection of intrathoracic airway trees. *Medical Imaging, IEEE Transactions on*, 15(3):314–326, 1996.

[18] R. M. Summers, D. H. Feng, S. M. Holland, M. C. Sneller, and J. H. Shelhamer. Virtual bronchoscopy: segmentation method for real-time display. *Radiology*, 200(3):857–862, 1996.

[19] O. Weinheimer, T. Achenbach, and C. Düber. Fully automated extraction of airways from ct scans based on self-adapting region growing. *Computerized Tomography*, 27(1):64–74, 2008.

Visual Impairment Simulation for Inclusive Interface Design

Veljko B. Petrović
University of Novi Sad—Faculty of Technical Sciences
Dositeja Obradovića 6
21000, Novi Sad, Serbia
pveljko@uns.ac.rs

Dragan Ivetić
University of Novi Sad—Faculty of Technical Sciences
Dositeja Obradovića 6
21000, Novi Sad, Serbia
ivetic@uns.ac.rs

ABSTRACT

This paper describes research in developing disability simulation used for inclusive design of user interfaces. It presents a medium-fidelity prototype which simulates visual impairment caused by Age-Related Macular Degeneration in real time on arbitrary user interfaces, and describes how the prototype design was arrived at. It does so by surveying previous work in the field, identifying broad trends, and systematizing problems visual impairment simulation systems must solve. This systematization focuses on issues of simulator portability, the importance of eye-tracking, the vital nature of real-time performance, the flexibility of the solution, and veracity of the simulator to actual AMD symptoms.

Keywords

HCI, AMD, macular, impairment, inclusive

1. INTRODUCTION

This paper describes the development of a disability simulation framework focusing on visual impairment, specifically visual impairment caused by maculopathy, common in disorders such as Age-Related Macular Degeneration (AMD). This framework is developed in order to support inclusive design, a term used to denote design focused on universal usability—allowing for maximal possible variability in users targeted by an interface. The importance of this is underlined by Shneiderman's eight golden rules of interface design being updated to include universal usability[Shn09a]

AMD was chosen as a subject of particular study because of how common it is and how much more common it is likely to become, given that it is a gerontological disorder, and the human lifespan is increasing[Uni02a]. It also—as shall be presented later—presents with a wide variety of symptoms, making it a useful test-case for considering the modeling and simulation of visual impairment in general.

The paper first presents the case that there is a problem with user interfaces (UI) and people with AMD, then that the problem is not worth dismissing, and finally that disability simulation is a valid approach to ameliorating that problem. The paper then presents previous work in this field, identifies certain trends and suggests, based on those trends, the need for a more comprehensive framework for disability simulation, using the AMD focus as both illustrative and alone worth the effort. A systematization of problems facing a visual impairment system some of which are addressed in previous work and some not, is presented, and then used to outline a visual impairment system a medium-fidelity prototype of which is then presented.

This paper is divided into seven sections: the first is the introduction, the second discusses the validity of the approach and previous work in the field, the third outlines the problems a general visual impairment framework must solve, the fourth outlines the software prototype implemented, the fifth outlines the conclusion and further avenues of research, the sixth contains the acknowledgements, and the seventh contains the references.

2. IMPAIRMENT AND SIMULATION

The first question that arises when considering this research is whether it represents a suitable investment of time and attention: is AMD (and by extension visual impairment) a big enough problem? The answer to this question follows from the nature of AMD as a disorder and its epidemiology.

AMD—Nature, Epidemiology, and Risk

A detailed aetiology of Age-Related macular degeneration is outside the scope of this paper, however, for purposes of orientation is suffices to say that AMD is a progressive degenerative disease of the macula, a region of the retina responsible for central (as opposed to peripheral) vision. It is divided

into ‘dry’ and ‘wet’ varieties, but for the purposes of designing interfaces there is no significant difference between the two. Its cause is unknown, and there is no effective treatment (though certain forms of surgery cure certain forms of ‘wet’ AMD by reversing malign neoplasia of retinal blood vessels).

Are there likely to be many users with AMD? The total number of computer users is increasing. This is especially true if all devices with interfaces are counted, such as phones, tablets, consoles, and smart TVs. As the number of users increases, it is expected that a significant part of the increase will be from the elderly, partially from increased market penetration and partially from the existing user-base growing older. This is inevitable as for some key technologies the market penetration is rapidly approaching 100%. According to the ITU[Int13a] the penetration rate for mobile phones is 96% world-wide, and the penetration rate for an online presence is 39%. These numbers become 100% and 77% respectively if only Europe is considered. As these numbers increase—as trends indicate—they will invariably include the elderly as well, especially since, according to the UN[Uni02a], the population of Europe is aging significantly. It is expected that the population of over-sixties will reach 28.8% by 2025.

One can establish a lower bound for the number of interface users in this demographic by looking at interface use proxies: the prevalence of social network use for people over 65 is 32%[Dug13a]The upper bound trends towards 100% as technological progress mandates the use of interfaces in order to have an independent life.

The prevalence of AMD is difficult to determine since diagnosing AMD is a nontrivial task, and it often goes unreported in its earlier stages. However, a number of studies have been done on the epidemiology of AMD, with markedly varied results. Less conservative estimates give such results as 64 % of people over eighty[DeJ06a] modulated by certain risk factors[Mar11a]. More conservative estimates broadly agree on lower but still worrying levels of prevalence, with results such as 11.90% for men over 80, and 16.39% for women over 80[Gro04a], or 3.7% for people between 75 and 84, and 11.0% for the total population over 85[Vin95a]. Either way, with the aging of the population being what it is, this prevalence is expected to double in the future and to increase by at least 50% by 2020[Gro04a].

If the more conservative figures are applied to the USA—serving here as a model nation due to easy access to detailed census data[Bur15a]—we find that according to [Gro04a] the estimated number of people with AMD is 1,658,000. The Rotterdam study, one the other hand, indicates results of an approximate total of 1,087,000. These approximate results indicate that just under 1% of the adult population of the United States has AMD, not counting earlier cases of the disease in the 55-70 range. Of course, as the population ages and life-extending medical care becomes more sophisticated, this number will increase.

It is evident, therefore, that there does exist a population of users with AMD. Does this population have a significant amount of difficulty when using interfaces? A systematization of the symptoms of AMD can be seen in section 3.1, but briefly, AMD leads to general loss of acuity, loss of color and contrast sensitivity, gaps in the visual field first visible in text, the loss of a central (foveal) sight (in whole or in part), and unpredictable shifting deformation of the visual field (metamorphopsia). This is a considerable amount of impairment and previous research in this field[Sco02a] shows conclusively that conventional interfaces are not suitable for people with AMD.

Disability Simulation and Interfaces

Disability simulation is the practice of creating some sort of apparatus which simulates the experience of having some sort of disability. Its original purpose was as an aid of empathy[Wil69a], but careful analysis shows that it is flawed in achieving this[Flo07a]. However, it can still be used to foster a rather more practical form of empathy—simulating a disability is a great way for a designer to gauge how a design will be perceived and used by people with disabilities. This can be the virtual modeling of users for the purposes of ergonomic design[Kak12a], the purposes of rehabilitation and accessibility design[Har14a], or for the purposes of UI design as in the Cambridge Impairment Simulator[Bis13a][Bis12a].

Of course, when doing usability testing nothing can possibly replace testing using people who actually have AMD (or other disabilities and disorders), but the use of disability simulation—occasionally also called user modeling—is crucial in allowing for the iterative testing of an interface. This iterative testing using simulation and approximation is crucial for what’s referred to as ‘inclusive design’ as opposed to designing the interface exclusively for the able-bodied and then adding accessibility features later. The utility of disability simulation is such that it was the focus of a Horizon 2020 FP7 EU project[Ver15a], which included visual impairment as well[Sul13a].

Thus, clearly, there does exist a significant problem and it is very likely that disability simulation is the way it can be at least ameliorated.

Previous Work

The idea of disability/impairment simulation is not new and has been explored in various settings for various applications. A survey of the literature has shown that previous work can be reasonably divided

into either application-specific simulators or universal attempts to simulate impairment. Application-specific simulators focus on one specific application either because they focus on researching one activity to the exclusion on others or because they deliberately reduce their focus to a specific platform to increase their ability to accurately simulate impairment.

The activities researched with application-specific simulators of the first kind are mostly those tasks that impact most heavily on independent life: reading and driving. Driving studies evaluate how well affected people can drive, and how much help visual aids are[Pel05a]. In the matter of reading there's been research on the eye-movements of the impaired[Pid06a] with applications in rehabilitation[Var04a] or in visual aid development[Har14a]. Likewise, application-specific simulators sometimes focus on the application platform such as Swing/NetBeans[Vot09a].

Attempts to simulate impairment for any sort of application are generally focused on acquiring the video output of GUI rendering and then modifying it in order to simulate the effects of impairment. Some of these are heavily hardware based, such as the case with parts of the Inclusive Design Toolkit[Inc15a] which relies on specially made glasses to simulate certain visual impairments, but most solutions are predominantly software-based. The most sustained work on this field is the work on the groundbreaking Cambridge Impairment Simulator [Bis13a][Bis12a][Goo07a] which is a vital part of the Inclusive Design Toolkit and the relevant perceptual model[Bis08a]. A number of tools have also been developed partially or fully outside of academia. These tools purport to help with inclusive design by simulating visual impairments. The most interesting of such projects are the Visual Impairment Simulator[Vis15a] and WebAIM Low Vision Simulator[Web15a].

Lastly, a few solutions do not fit these categories: Some research has been done in using simulations to evaluate the severity of various impairments from a medical point of view[Fin99a], and there is also work on visual field simulation which touches on the subject of visual impairment simulation but focuses, instead, on optimal resolution for gaze-contingent displays[Per02a].

The solutions analyzed are equally heterogeneous in their means and their ends. One quarter use an analogue system for vision alteration, relying on specialized lenses that deform the user's visual field. The rest rely on active simulation, either using software tools (66.67%) or specialized hardware (8.33%) of those, 33.33% are gaze-contingent, and the rest (36.36%) either ignore gaze or use a gaze-proxy. Also heterogeneous are the fields and ultimate goals of the solutions: 41.67% are fundamentally ophthalmological in purpose, half are intended to aid inclusive design, and 8.33% are special purpose.

Each solution succeeds on its own terms, resolving those problems the authors intended to tackle. However, as is the case in any research there are still open questions to be addressed. One of the key things to consider with all of these solutions is that they pick and choose which symptoms they simulate and to which extent. In certain cases, such as in [Har14a] or [Per02a] this is clearly a deliberate choice because only some factors were of interest to the authors. In other cases the choice is not deliberate, but is instead a unwanted but necessary compromise with technological limitations. Either way, it is necessary to acknowledge the limits of what was simulated in order to be able to ascertain the applicability of the simulation to actual design work.

3. OPEN QUESTIONS

This section deals with the open questions left after the previous work, especially those whose answer pertains to the development of a general framework for visual impairment simulation. AMD is used as a test-case because it is significant, sufficiently frequent, and presents with a wide array of symptoms. The questions to be answered can be organized into questions of:

- veracity,
- performance,
- universal applicability, and
- scalability.

It should be pointed out that these are not questions *entirely* unaddressed in previous work. Rather, their central nature is such that, even when they have been addressed, further work is necessary. In brief, veracity means that the framework must replicate the impairment as accurately as it is possible, performance means the framework must allow simulators to run in real-time, universal applicability means that the framework must allow for a wide selection of target interfaces, and scalability means that the framework must be accessible as simply as possible to as large an amount of interface designers as possible regardless of budget.

Veracity

It is not immediately obvious why veracity is important. It is quite reasonable to say that it is only necessary to simulate the 'important' symptoms of a visual impairment while leaving the others out. The difficulty, of course, is to determine what 'important' is for the purposes of interface design. To assume what is important to an interface is to ignore the perspective of the visually impaired—the exact same empathy deficiency disability simulation was created to solve[Wil69a].

Previous work clearly addresses this question, but does so in a haphazard fashion—not due to incompetence or oversight, but due to different focus. Not one of the surveyed solutions, for instance, implemented metamorphopsia, and most focused on the most obvious symptom: the central scotoma.

It is, however, easy to say that the simulation must be faithful to the impairment it seeks to emulate. It is quite more difficult to say how such a thing may be done. Using AMD as an example the first step is to gather the symptoms as they are described in the medical literature:

a) The user may experience reduced general visual acuity[Sco02a]
b) The user may experience reduced ability to perceive color correctly[Sco02a]
c) The user may experience a difficulty[Sco02a] discriminating between similar light levels in a picture as measured by the Pelli-Robson Contrast Sensitivity Chart.
d) The user may experience minor gaps in their visual field causing letters to drop out of text[Dej06a] or for lettering on densely-formatted documents to seem misaligned causing problems in, e.g., reading tables.
e) The user may experience more significant foveal (central sight) scotoma (gaps in the visual field), blocking portions of the visual field[Dej06a]. These gaps may be visible as voids, spots, or deformations. Voids are filled in by the visual cortex in the same way the scotoma caused by the optic nerve is, spots are visibly dark or black, and deformations are visibly flickering as in the case of the scintillation scotoma or in some way distorted images which block part of the visual field.
f) The user may experience a complete loss of central sight[Dej06a].
g) The user may experience significant metamorphopsia—a deformation of the visual field which causes straight lines to appear curved and shifting[Dej06a][Rio08a] and causes visual elements to appear misaligned.
h) The user may experience complete (for legal purposes) loss of vision[Dej06a][Rio08a].

Once the symptoms are gathered it is tempting to provide ad-hoc implementations for all of them. However principles of good design, not to mention the sheer number of possible impairments preclude this approach. While the development of a full visual impairment modeling language is beyond the scope of this paper—though one is being developed—the simplest way to understand symptoms of visual impairments from the point of view of the simulator/framework designer is to divide them into the selector and effector components. Selectors determine which part of the visual field is affected and can be composited from such components as: the whole field, vision of the fovea, vision of the foveola, peripheral vision, random subsections, and text, where compositing is done using simple set intersection. Effectors control how the selected areas of the visual field are modified. One possible way to systematize such changes is to base them on visual variables.

Variable	Symptoms
Position	(a)(d)(e)(f)(g)
Size	(a)(g)
Shape	(a)(g)
Value	(c)
Color	(b)
Orientation	(g)
Texture	(a)(d)(c)(g)

Table 1. Mapping symptoms to visual variables.

Visual variables[Ber83a][Gar09a] are a system of describing various ways in which an image informs the viewer. Originally intended as a way of systematizing and discussing cartography, they were later adapted to various other problems including interfaces and visualization[Car03a]. The visual variables are: position, size, shape, value, color, orientation, and texture. Table 1 schematizes the connection between variables and AMD symptoms for purposes of illustration.

These connections are useful in the broader context of developing a universal approach to disability simulation and modeling, as the changes made by the impairment to certain subsections of the visual field can be explained in terms of effectors corresponding to visual variables, changing, say, position, or value, or texture or some combination thereof.

It should be noted that appropriately simulating most of these symptoms demands discriminating between central vision and peripheral vision which necessitates both some way of tracking the user's gaze and knowing the distance between the user and the display. Distance is necessary because the description of the visual field must be in terms of degrees of the visual field. Converting this into pixels demands the distance from the display. Not all of the previous proposed solutions consider this, with only those designed for ophthalmological purposes paying much attention. This issue is further discussed in the subsection on scalability.

Performance

The first question regarding performance is to ask if it is necessary at all. Aside from the obvious rejoinder that no piece of software is better if it is slower, it should be said that real-time simulation allows for a piece of software to be used in a way that closely mimics the way a visually impaired person might use it. Offline simulation based on video might be useful, but will never allow for testing protocols or traditional usability evaluation. In previous work, some efforts were offline, some didn't use software processing at all, relying on optics to simulate disability, and others can be divided into those which used slow intercepts (100ms times have been reported in [Vot09a]) or those which operated quickly, but had limited capabilities, such as solutions based on hardware overlays.

When it comes to performance, only two significant problems present themselves. The first is the problem of text-based selectors. While it is possible to completely avoid its use, this is only feasible through very precise gaze-tracking with a very high sampling frequency—enough to fully capture and subvert saccade movements—which is not always practical, as is discussed in the subsection on scalability. In case text-based selectors are used, this necessitates some way to recognize text using computer vision algorithms. Current algorithms meant for real-time text recognition run in timeframes around 300ms[Neu12a], but is likely that using a simplified method—specifically stopping at stage one classification—some increase in performance could be possible. The goal, of course, is to have sub 30ms times in order to allow for 30fps functioning of the impairment simulator.

The second problem depends on how the simulator gets the image of the interface it plans to deform. There are three approaches: Toolkit level intercept, compositing level intercept, and raster level intercept. Toolkit level intercepts are out of the question because this will fail to answer the question of universal applicability. If the image capture is done by relying on the toolkit used to generate the GUI, then interfaces done using any other type of toolkit are impossible to simulate. Compositing level intercept is better—this attempts to capture DirectX or OpenGL commands a piece of software is sending to the driver, and uses those to capture footage. Normally this is used to record footage of 3D applications and video games. Unfortunately this approach is unlikely to work with normal 2D Windows applications and is not cross-platform at all.

In practice the best two approaches—on Microsoft Windows, which was chosen in order to support as many developers as possible—are DirectX front surface readback and direct read of the screen buffer using the bitblit Win32 function. Of these two, the latter has shown to be slightly faster and delivers steady 30fps in most cases, though it struggles to go much past that that.

Universal Applicability

While a simulator would be much easier to construct using laboratory grade equipment, high-end hardware, and precisely controlled circumstances, this rather defeats the purpose of inclusive design. Inclusive design is meant to be universal, as there's no telling which interface element of which software package will be used by a visually impaired person. To allow for this, the simulator must be accessible to everyone, no matter their software or hardware, and it must be such that it does not place any undue burden on the user who should focus on the interface design above all.

It is doubtless true that better hardware allows for better simulations: higher fidelity and higher efficiency leading to better results. However, such hardware is hard to come by and expensive, and accessibility and inclusive design are already a low priority in a lot of commercial software. Adding a hefty price tag does not help inclusive design becoming a universal in UI engineering, rather the opposite. Thus, it is crucial that the simulator be capable of running in situations with little to no specialized hardware, adapting to limits in accuracy as best it can. Naturally, in the presence of suitable hardware it can adapt to utilize those superior resources increasing its efficiency. However, it cannot demand such hardware be present without jeopardizing its goal of propagating inclusive design. This requirement for adapting to changing circumstances is outlined further as a part of scalability.

Scalability

The scalability requirement combines affordability and ease of use—it represents what is required to allow the framework to reach ubiquitous use. The two key goals here are plug-and-play installation and no need for specialized hardware. This latter goal is the most difficult one because veracity demands gaze-tracking in order to differentiate between peripheral and central vision which need to be treated markedly differently even in healthy adults[Per02a]. Since the presence of purpose-built gaze-tracking hardware cannot be relied upon, some alternative solution needs to be found. The two approaches that present themselves are tracking a proxy for the user's gaze or implementing a gaze-tracking solution which uses hardware that *can* be relied upon, such as a webcam.

The proxy used for the user's gaze in the literature is naturally the position of the mouse cursor, however, this poses difficulties which may be impossible to

resolve. The idea is that the tester or developer will be instructed to keep his or her eyes focused on the position of the cursor throughout testing thus obviating the need for accurate gaze-tracking. The problem there is the scenario where the user has, say, a simulated foveal scotoma obscuring a vital part of the interface forcing the user to improvise using peripheral vision. Will the user be so disciplined to avoid a few quick—barely liminal—glances with his or her central vision?

While this problem could be studied separately, it is actually possible to get a good idea by consulting a field distant from HCI. Averted vision is a venerable technique of observation in astronomy[Bar77a] and it consists of doing just what is expected of the user in the gaze proxy solution: Keeping visual focus on some other object and using peripheral vision to observe the target. Considering that averted observation was—and is—considered something which requires training and which is easy to do wrong, it can be assumed that using essentially the same approach in visual impairment simulation is equally difficult. Further, off-center focusing is a skill that helps people adapt to scotomata and it has been determined that even people with an accurate simulation of a foveal scotoma can only be trained to avert their gaze correctly after five hours of training[Har14a].

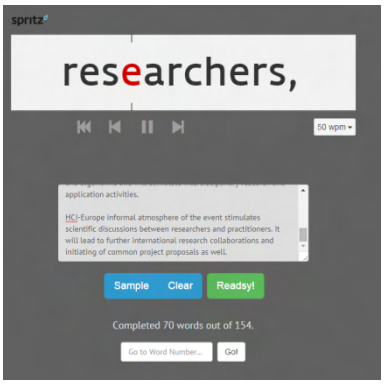


Figure 1 Original unmodified interface for RVSP

One alternative is to implement a webcam-based eye-tracking solution. This is difficult: commercial eye-tracking solutions generally use near-IR sources to illuminate the eye which is then recorded using a high-FPS camera. However, the problem is made easier when it is considered that the area of central vision is between 3° and 13° depending on which acuity threshold one wishes to adopt as the 'edge' of central vision—features of human being rarely yield to sharp distinctions. The size of 1-5° for a foveal scotoma is attested in the literature. Thus the system need only be accurate enough to capture a region of interest of that size, no smaller, which simplifies matters.

Figure 2 RVSP interface modified by medium-fidelity prototype of impairment simulation

While the technology to use webcams to track the user's gaze does exist[Sew10a] it is not equal to IR-based systems[Bur14a]. Commercially available systems boast accuracy rates of around 1.7°[Sti15a], but suffer issues due to lack of lock and sensitivity to light.

Another possible solution is to use a gaze proxy like cursor position, but to track user distance and to rigorously simulate the visual field and, crucially, the difference in acuity between peripheral and central vision. This ameliorates the problem of quick barely liminal glances outlined above. This is less veracious than the webcam approach, but is maximally scalable, especially since a webcam based solution may cause technical glitches because of jitter and drift, while one using this enforced-proxy approach has no such issues.

4. SOFTWARE PROTOTYPE

The methodology to tackle these problems is to develop a universal software simulator of visual impairment which seeks to better answer the four open questions outlined above.

As a testbed for further development a software prototype was built which simulates all the symptoms of AMD: scotoma, loss of central sight, metamorphopsia, loss of acuity, and loss of contrast and color perception. A complete loss of all vision was not simulated. This helped increase veracity: the ability to simulate acuity and contrast problems helped 'hide' the noncentral scotomata: creating an effect which corresponds to what patients report in the literature where the dropping out of parts of the visual field can be imperceptible while still creating problems. Further, the use of simulated metamorphopsia helped illuminate problems with relying on component alignment in UI design.

The prototype used DirectX front surface readback and bitblit-based raster read of the screen buffer, and DirectX 9.0c for the rendering of the changed image. It then used DirectX to apply all the changes to the image, using a SM4 pixel shader to implement all

effects except metamorphopsia which was implemented through render geometry deformation via a vertex shader. All of the effects are parameterized and can be tuned or entirely disabled depending on the severity of the AMD simulated. Eventually, this parameterization will come directly from an impairment model and allow for greater granularity. This approach was optimized until, with the use of bitblt (which proved faster in-implementation than buffer readback) and shader model 4 implementations of symptoms, a fixed framerate of approximately 30fps was achieved even during system load. Stress tests were performed using simulated CPU loads and Unreal Engine 4 authoring tools which served as a stand-in for graphically demanding applications.

No specialized hardware is required for the implementation of this testbed prototype, much in the same way that further improvements will not require any specialized hardware either. A simple development workstation is sufficient to run the software and benefit from its simulation. This makes it universally accessible: any developer can run it on the same machine used to design the interface in the first place and, thus, has little excuse not to do so. Scalability is achieved by adapting to any proxy the user's gaze available. In case of the presence of a gaze-tracking device, all that needs to change is that the symptoms are no longer calculated from the cursor position.

Figure 1 shows the unmodified original interface, and Figure 2 shows the simulator working. The prototype used a user gaze proxy based on the position of the mouse cursor while future versions will also support commercial eye-trackers and webcam eye tracking.

5. CONCLUSION

It is both possible and desirable to employ visual impairment simulation in interface design. Previous work in the field shows that there is a need for this sort of software, that such software can be made, and that such software can be made better. Or, rather, can be made in such a way as to incorporate various good features of several approaches in order to minimize wasted effort and allow the designers to easily come to understand the needs of all of their users.

Inclusive design can no longer remain an option, not when the ability to use an UI, whether fitted to a computer, a phone, a television, or a voting machine is a prerequisite for any level of participation in life and the economy. New tools and approaches will have to be developed in order to achieve this, and disability simulation is one step forward.

This paper demonstrated the need for visual impairment simulation, indicated and systematized the questions any framework for such simulation must answer, and offered tools for modeling such solutions by using visual variables as language for describing alterations caused by impairment. It also provided a medium-fidelity prototype of such a solution.

This research opened up several possible future avenues of research including the full design of a framework partially specified in this paper, and a design for a language for specifying visual impairments. Further, the precise efficacy of webcam-based gaze-tracking will have to be established for this particular application and an approach that's maximally tolerant to changes in lightning conditions, motions of the head, and poor calibration will have to be developed. The presence of a stable light-source and of a trained operator cannot be relied upon if the goal is, as it should be, the universal acceptance of inclusive design as the 'new normal.' Thus, the current state of the art for webcam based eye-tracking is insufficient for the needs of visual impairment simulation. Either the state of the art will have to be improved, or a greater tolerance to problems will have to be built into the simulator solution.

6. ACKNOWLEDGMENTS

This work is financially supported by Ministry of Science and Technological Development, Republic of Serbia; under the project number TR32044. "Development of software tools for the analysis and improvement of business processes", 2011-2014.

7. REFERENCES

[Bar77a] Barrett, AA. Notes-Aristotle and Averted Vision. Journal of the Royal Astronomical Society of Canada 71, pp.327, 1977.

[Ber83a] Bertin, Jacques. Semiology of Graphics: Diagrams, Networks, Maps, 1983.

[Bis13a] Biswas, Pradipta, and Pat Langdon. Inclusive User Modeling and Simulation. A Multimodal End-2-End Approach to Accessible Computing, pp.71–89, 2013.

[Bis12a] Biswas, Pradipta, Peter Robinson, and Patrick Langdon. Designing Inclusive Interfaces Through User Modeling and Simulation. International Journal of Human-Computer Interaction 28. pp1–33, 2012.

[Bis08a] Biswas, Pradipta, Tevfik Metin Sezgin, and Peter Robinson. 2008. Perception Model for People with Visual Impairments. Visual Information Systems, Web-Based Visual Information Search and Management, pp.279–90, 2008.

[Bur15a] Bureau, U. S. Census. American FactFinder – Results, 2015.

[Bur14a] Burton, Liz, William Albert, and Mark Flynn. A Comparison of the Performance of Webcam vs. Infrared Eye Tracking Technology. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 58 pp.1437–41, 2014.

[Car03a] Carpendale, MST. Considering Visual Variables as a Basis for Information Visualisation. Computer Science TR# pp.2001-693, 2003.

[Dej06a] de Jong, Paulus T.V.M. Age-Related Macular Degeneration. New England Journal of Medicine 355 pp.1474–85. doi:10, 2006.

[Dug13a] Duggan, Maeve, and Joanna Brenner. The Demographics of Social Media Users, 2012. Vol. 14. Pew Research Center's Internet & American Life Project, 2013.

[Fin99a] Fine, Elisabeth M, and Gary S Rubin. Effects of Cataract and Scotoma on Visual Acuity. Optometry and Vision Science 76, 1999.

[Flo07a] Flower, Ashley, Matthew K. Burns, and Nicole A. Bottsford-Miller. Meta-Analysis of Disability Simulation Research. Remedial and Special Education 28 pp72–79, 2007.

[Gar09a] Garlandini, Simone, and Sara Irina Fabrikant. Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization. Spatial Information Theory, pp. 195–211. Springer, 2009.

[Goo07a] Goodman-Deane, Joy, Patrick M. Langdon, P. John Clarkson, Nicholas HM Caldwell, and Ahmed M. Sarhan. Equipping Designers by Simulating the Effects of Visual and Hearing Impairments. Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility.ACM.pp241–42, 2007.

[Gro04a] Group, Eye Diseases Prevalence Research, and others. Prevalence of Age-Related Macular Degeneration in the United States. Archives of Ophthalmology 122. pp. 564, 2004.

[Har14a] Harvey, Hannah, and Robin Walker. Reading with Peripheral Vision: A Comparison of Reading Dynamic Scrolling and Static Text with a Simulated Central Scotoma. Vision Research 98. pp54–60, 2014.

[Inc15a] Inclusive Design Toolkit Home: http://www.inclusivedesigntoolkit.com/betterdesign2/, 2015.

[Int13a] International Telecommunications Union. World Telecommunication/ICT Indicators Database 17th Edition,2013.

[Kak12a] Kaklanis, Nikolaos, Panagiotis Moschonas, Konstantinos Moustakas, and Dimitrios Tzovaras. Virtual User Models for the Elderly and Disabled for Automatic Simulated Accessibility and Ergonomy Evaluation of Designs. Universal Access in the Information Society 12. pp.403–25, 2012.

[Mar11a] Mares, Julie A, Rick P Voland, Sherie A Sondel, Amy E Millen, Tara LaRowe, Suzen M Moeller, Mike L Klein, et al. Healthy Lifestyles Related to Subsequent Prevalence of Age-Related Macular DegenerationHealthy Lifestyles and Prevalence of AMD. Archives of Ophthalmology 129. pp.470–80, 2011.

[Neu12a] Neumann, L., and J. Matas. Real-Time Scene Text Localization and Recognition. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp.3538–45, 2012.

[Pel05a] Peli, E., A. Bowers, A. Mandel, K. Higgins, R. Goldstein, and L. Bobrow. Design for Simulator Performance Evaluations of Driving with Vision Impairments and Visual Aids. Transportation Research Record: Journal of the Transportation Research Board 1937. pp.128–35. 2005.

[Per02a] Perry, Jeffrey S., and Wilson S. Geisler. Gaze-Contingent Real-Time Simulation of Arbitrary Visual Fields, 2002.

[Pid06a] Pidcoe, P. E. Oculomotor Tracking Strategy in Normal Subjects with and without Simulated Scotoma. Investigative Ophthalmology & Visual Science 47. pp.169–78, 2006.

[Rio08a] Riordan-Eva, Paul, and John Whitcher. Vaughan & Asbury's General Ophthalmology, 2008.

[Sco02a] Scott, Ingrid U, William J Feuer, and Julie A Jacko. Impact of Graphical User Interface Screen Features on Computer Task Accuracy and Speed in a Cohort of Patients with Age-Related Macular Degeneration. American Journal of Ophthalmology 134. pp.857–62, 2002.

[Sew10a] Sewell, Weston, and Oleg Komogortsev. Real-Time Eye Gaze Tracking with an Unmodified Commodity Webcam Employing a Neural Network. CHI '10 Extended Abstracts on Human Factors in Computing Systems. pp.3739–44, 2010.

[Shn09a] Shneiderman, Ben, Catherine Plaisant, Maxine Cohen, and Steven Jacobs. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 5th ed. Pearson, 2009.

[Sti15a] Sticky: http://www.sticky.ad/.

[Sul13a] Sulzmann, Frank, Roland Blach, and Manfred Dangelmaier. An Integration Framework for Motion and Visually Impaired Virtual Humans in Interactive Immersive Environments. Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life. pp.107–15, 2013.

[Uni02a] United Nations Department of Economic and Social Affairs Population Division. World Population Ageing: 1950-2050, 2002.

[Var04a] Varsori, Michael, Angelica Perez-Fornos, Avinoam B. Safran, and Andrew R. Whatham. Development of a Viewing Strategy during Adaptation to an Artificial Central Scotoma. Vision Research 44. pp.2691–2705, 2004.

[Ver15a] VERITAS FP7 IP: http://veritas-project.eu/index.html.

[Vin95a] Vingerling, Johannes R, Ida Dielemans, Albert Hofman, Diederick E Grobbee, Michel Hijmering, Constantijn FL Kramer, and Paulus TVM de Jong. The Prevalence of Age-Related Maculopathy in the Rotterdam Study. Ophthalmology 102. pp.205–10, 1995.

[Vis15a] Visual Impairment Simulator for Microsoft Windows: Visual Impairment Simulator for Microsoft Windows: http://vis.cita.uiuc.edu/.

[Vot09a] Votis, K., T. Oikonomou, P. Korn, D. Tzovaras, and S. Likothanassis. A Visual Impaired Simulator to Achieve Embedded Accessibility Designs. IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009. pp.368–72, 2009.

[Web15a] WebAIM: Low Vision Simulation: http://webaim.org/simulations/lowvision.

[Wil69a] Wilson, Earl D., and Dewaine Alcorn Disability Simulation and Development of Attitudes toward the Exceptional. The Journal of Special Education 3. pp.303–7, 1969.

Multiphase Action Representation for Online Classification of Motion Capture Data

Samer Salamah
Faculty of Computer Science, GDV Chemnitz University of Technology
Str. der Nationen 62
09111, Chemnitz, Germany
samer.salamah@s2008.tu-chemnitz.de

Guido Brunnett
Faculty of Computer Science, GDV Chemnitz University of Technology
Str. der Nationen 62
09111, Chemnitz, Germany
guido.brunnett@informatik.tu-chemnitz.de

ABSTRACT

In this paper we introduce a novel, simple, and efficient method for human action recognition based on a multiphase representation of human motion. An action is considered as a finite state machine where each state represents a primitive motion called motion phase, which is simply a sequence of poses with predefined common features. Spatial-temporal and postural features introduced in previous work are redefined by using only 3D joint positions for features extraction and are extended by involving the relative movement of the body end-effectors as new features. We developed a framework for modelling a given motion in the proposed motion model, whereupon we used this framework to create a model database of 25 different actions. Using this database we conducted a number of experiments on data obtained from several sources as well as on distorted data. The results showed that the presented method has high accuracy and efficiency. Additionally, it can work offline and online in real time, and can be easily adapted to work on 2D data.

Keywords

Human motion, motion capture, motion segmentation, motion classification, action recognition.

1. INTRODUCTION

Motion capture data is the basis for a realistic animation, but it is expensive to produce, therefore, the reusability of it is very important. However, this reusability demands that the motion capture data is good segmented and annotated. The segmentation into natural motion phases increases the reusability; however, the basis for this segmentation is the recognition of motion phases. Moreover, motion capture data is used in medicine for the analysis and examination of joint movement and rehabilitation procedures. These fields continuously produce large stores of data so that it is hard and tedious to retrieve a particular motion manually. Therefore, many methods have been developed for automatic search and retrieval in these stores. Of late, marker-less motion capture data has achieved significant improvement in accuracy, which enables it to be used in control and surveillance systems, as well as in the human–robot interaction field. This demands instantaneous and precise action recognition, which is what our presented method can do. Many works such as [Jin07a] and [Bar04a] successfully could reduce the high dimensionality of motion capture data without semantic lost. Additionally, some other works such as [Liu06a] and [Zha11a] could capture meaningful human motion with a reduced marker set. Inspired by such works, we develop a motion model that depends only on the movement of the actor's end-effectors and some basic postural features. We extend the features introduced in [Sal15a] so that any primitive motion can be described automatically in high-level terms. In general an action consists of several phases each of which is represented by a subset of these features and characteristics. Using the framework of phases and features a person with no experience with motion capture data is able to define or design movements at will and use them in any application area to retrieve and classify motions from motion repositories or to recognize ongoing motions online in real time.

The contribution of the proposed method is threefold: (1) specification of high-level features of human motion that enables (2) multiphase representation of human action and (3) utilizing this framework for efficient and high-accuracy classification of motion capture data. The presented approach is easy to implement, efficient, and works in real time both online and offline. Additionally, the database of recognizable motions can be extended easily in a very short time because there is no need for training data or training time. The rest of this paper is organized as follows: First, an overview of the

related works is given, and then some terms and notations used in our work are introduced. After that, the proposed features are described in Section 4 while the developed motion model is introduced in Section 5. In Section 6 the classification algorithm is presented, and then in Section 7 some conducted experiments are described and their results discussed. Finally, the work is concluded in Section 8.

2. RELATED WORK

Action recognition from motion capture data has received a lot of attention in the last decade. Nowadays there is a wide range of methods for classification of motion capture data. These methods can be divided into online and offline methods depending on whether the whole data should be processed before a classification result can be given or not. From another point of view, the classification methods can be divided into the following groups based on the nature of the features used to represent human motion as well as the field in which the used algorithms originated:

Description-Based

Methods of this category use annotated motion templates and high-level semantic features for action recognition. The work of J Baumann et al. [Bau14a] is an example of these approaches, where a motion capture database is annotated with actions of interest in an offline phase, and then used in the online phase to search for motion segments that are similar to annotated actions in the motion database. Leightley et al. [Lei14a] used Exponential Map EMP and k-means clustering to model human actions. For each action class they transform each pose of a representative sequence into EMP form then they used k-means clustering to extract a small number of exemplars that represent the action. Then they used Dynamic Time Warping and Template Matching to recognize actions from motion capture data streams.

Machine Learning

Machine learning techniques are widely used to classify 2D and 3D human motion. Cho and Chen [Cho13a] generated features for each motion frame based on the relative positions of joints, temporal differences, and normalized trajectories of motion. They then used them in training deep neural networks that they later used to classify motion capture data. Coppola et al. [Cop15a] extended the 3D Qualitative Trajectory Calculus (QTC3D) and used them to model human actions. Then they learned HMM to recognise human actions.

Statistics-Based

Statistical techniques such as Gaussian-Mixture-Models, Histograms and Space-Time Correlation are used here to model and recognize human motion. Y Jin and B Prabhakaran [Jin07a] quantized human motion data by extracting spatial–temporal features using SVD and then translated them into a one-dimensional sequential representation through a semantic Gaussian Mixture Models with Expectation-Maximization algorithm. These could reduce the dimensions of human motion data while maintaining semantically important features. M Zhang and A Sawchuk [Zha12a] introduced a framework for human motion modelling and recognition based on a bag of features. They modelled human activities through histograms of primitive symbols on physical features using k-means clustering and soft weighting. Unlike our proposed method, most of the above-mentioned methods are unable to separate two consecutive occurrences of one motion. In addition, the transitions between two motions are not recognized as transition but merged with the neighbour motions. Moreover, in some methods the learning process by classification is not simple, while our method is simple, easy to implement, efficient, and does not need any training phase.

3. PRELIMINARIES

We describe a pose of the human body as a set of annotated 3D points that correspond to the body joints. Thus, the human body pose is determined by the global 3D positions of these joints additional to the global orientation of the body. The proposed method needs a minimum set of joints *J*, namely, the ankles, knees, hips, chest, head, wrists, as well as a virtual joint at the pelvis called 'root'. In this work, we refer to ankles and wrists as feet and hands respectively. A pose at time *t* is described by $P^t = (q^t, p_1^t, p_2^t, \dots, p_n^t)$, where q^t is the global orientation of the body and p_j^t is the 3D global position of the joint *j*, where *n* is the number of used joints. The global body orientation at time *t* is given by three orthogonal vectors f^t, s^t, and h^t representing the normal vectors of the frontal, sagittal, and traversal main body planes respectively. We denote the single position coordinates of joint *j* at time *t* as x_j^t, y_j^t and z_j^t respectively where y_j^t is the vertical coordinate. We refer to the vector that goes from joint *a* to joint *b* at time *t* as $v_{a,b} = p_b^t - p_a^t$, and the motion direction of joint *j* at time *t* as $d_j^t = p_j^t - p_j^{t-1}$. Additionally, we define the motion magnitude of joint *j* at time *t* in the direction *v* as following $d_{j,v}^t = ||d_j^t|| \cos(\angle(d_j^t, v))$, where $v \in \{f^t, s^t, h^t\}$, and we refer to the algebraic sum $\sum_{t=s}^{t=e} d_{j,v}^t$ as the accumulated motion magnitude of joint j over the time interval $T = [s, e]$ in the direction *v*.

4. FEATURE DESCRIPTION

The main idea of the proposed method is based on a set of features that was inspired by the way in which people in general and kinesiologists in particular analyse and evaluate human motion. The method also

seeks to analyse the most important factors in deciding on the motion class. We extend the taxonomy tree of human motion introduced in [Sal15a] by adding motion directions of the end-effectors in the main body planes. The extended tree shown in Fig. 1 now consists of nine levels that reflect the importance of each group and the relations among features where the features in the first level have the highest importance. We call each complete path in this tree a 'pose state', which can be described as a complete set of the defined features. Each given pose is assigned a pose state by taking a previous pose into account. In the following, we introduce a detailed description of each of the features. In [Sal15a] the used features are calculated using both joint angles and 3D joint positions. However, we use here only 3D joint positions for calculating the introduced features.

Spatial–Temporal Features

In this section we introduce features that are generated by changing the joint positions over time, thereby denoting it as spatial–temporal features. They are introduced in the following in the order in which they are computed.

4.1.1 Motion Existence

First the existence of motion is checked. A pose is classified as dynamic if there is at least one joint that has moved a significant distance on at least one coordinate axis (1), otherwise it is classified as static.

$$\exists j \in J: \exists c \in \{x, y, z\}: \mid c_j^t - c_j^{t-1} \mid > \varepsilon \quad (1)$$

The threshold ε is a small real value representing the maximal noise value in the used data. Assuming there is a clip of *n* static poses that can be recorded during the system setup; the threshold ε is then the maximal displacement that a joint has achieved along any of the coordinate axes between two subsequent poses over the whole clip (2).

$$\varepsilon = \max_{t,j,c}(|c_j^t - c_j^{t-1}|) \text{ for all } t \in [2, n], \text{ all } j \in J \text{ and all } c \in \{x, y, z\}. \quad (2)$$

4.1.2 Motion Directions

Secondly, the motions of the end-effectors in the three main body planes are described. Based on the observation that almost all human actions are performed by displacing the body end-effectors, namely the hands, the feet, and the head/torso, we use the motion direction of these body parts as high-level features such as left foot moves forward up, or right arm moves left down fast. From a kinesiological perspective, the movements of body parts occur mainly in three anatomical planes, namely the frontal, sagittal, and traversal planes [Ham02a, Gre05a]. Based on this division of the body into three planes we define the directions of the joint movements relative to the body's axes as shown in Table 1.

Body Axis	frontal	vertical	sagittal
Positive Motion	forward	upward	left
Negative Motion	backward	downward	right

Table 1: Defined motion directions relative to main body's axes

4.1.3 Motion Space

Although the human body can move in many different ways, there are actually two major kinds of movements. These are locomotive, translator or linear, and non-locomotive, rotary, or angular [Ham02a, Gre05a]. If the whole body moves from one place to another, then the movement is locomotive; otherwise, it is considered as non-locomotive. A given pose is classified as locomotive if the root and both feet move, relative to the previous pose, in the same direction (3), or the root and at least one foot move in the same direction (4 and 5), while the other foot is fixed, and the accumulated magnitude of the root motion in the considered direction is greater than a certain threshold equal to the tibia length.

$$(\|d_{root}^t\| > \varepsilon) \wedge (\|d_{lfoot}^t\| > \varepsilon) \wedge (\|d_{rfoot}^t\| > \varepsilon) \wedge (d_{root}^t \cdot d_{lfoot}^t > 0) \wedge (d_{root}^t \cdot d_{rfoot}^t > 0) \quad (3)$$

$$(\|d_{root}^t\| > \varepsilon) \wedge (\|d_{lfoot}^t\| > \varepsilon) \wedge (\|d_{rfoot}^t\| \leq \varepsilon) \wedge (d_{root}^t \cdot d_{lfoot}^t > 0) \quad (4)$$

$$(\|d_{root}^t\| > \varepsilon) \wedge (\|d_{lfoot}^t\| \leq \varepsilon) \wedge (\|d_{rfoot}^t\| > \varepsilon) \wedge (d_{root}^t \cdot d_{rfoot}^t > 0) \quad (5)$$

where ε is the noise threshold defined in (2).

Postural Features

An important factor for classifying human motion is the change in the main body posture. We utilize this observation and use the following major and corresponding minor postures as features for the recognition of human actions.

4.1.4 Standing

In general, 'standing' is a major posture where the body maintains an upright position supported by the feet. The presented approach restricts the upright constraint to the lower body. Therefore, a pose is considered as 'standing' if at least one leg is extended and has a certain maximum inclination (6).

$$\left(\|v_{lfoot,lhip}\| > \eta\right) \wedge \left(\angle\left(v_{lfoot,lhip}, OY\right) \leq \alpha\right)) \vee \left(\|v_{rfoot,rhip}\| > \eta\right) \wedge \left(\angle\left(v_{rfoot,rhip}, OY\right) \leq \alpha\right)) \quad (6)$$

We consider a leg as extended if the distance between the foot and hip is greater than η, which is equal to one and a half of the femur length.

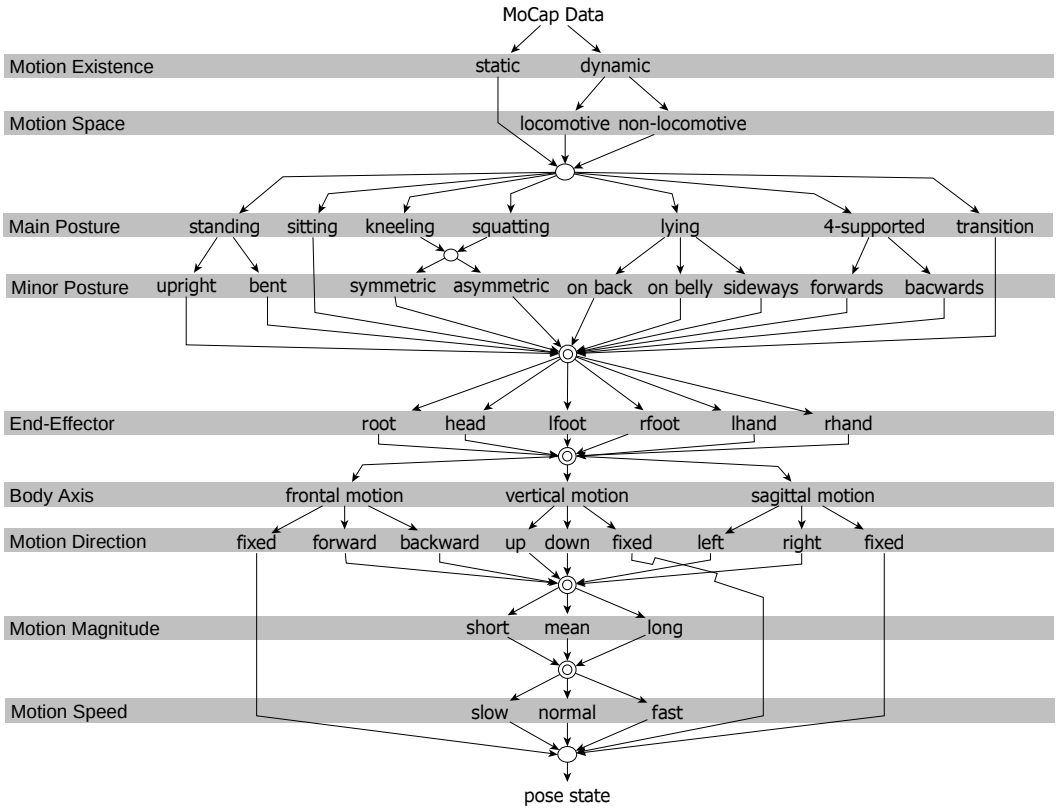


Figure 1: Taxonomy tree of human motion. Double circles allow the path to return to the first previous double circle, whereby it is not allowed to take the same path segment again.

The maximum inclination used by our experiments is $\alpha = 45°$. Standing can also have one of the following three minor postures:

1. If the torso stays upright, i.e. it has an inclination smaller than threshold β: $\angle(v_{root,cheast}, OY) \leq \beta$ (7), then the pose is considered as 'standing upright'. We used $\beta = 30°$.
2. Otherwise it is considered as 'standing bent': $\angle(v_{root,cheast}, OY) > \beta$. (8)
3. If the body is not supported only by the feet, then the pose is considered as 'standing leaned'. Suppose *S* is the set of support body parts, then 'standing leaned' is recognized when *S* contains at least one part except the feet $S \setminus \{p^t_{lfoot}, p^t_{rfoot}\} \neq \emptyset$. This minor posture, however, is in our case not recognizable, because motion capture data does not contain any information about the environment.

4.1.5 Sitting

The 'sitting' posture is a major posture in which the body is supported mainly by the buttocks rather than the feet, that implies that the projection of the gravity centre of the body lies outside the support base of the body formed through the feet. Additionally, the torso is not horizontal. Based on the height of the hip joint, it is decided whether the pose is sitting on an object or on the floor as minor postures. No constraints are put on the legs because there are many variants of the sitting posture according to the position of the legs. Legs can be vertical, crossed, or on each other.

4.1.6 Kneeling

'Kneeling' is also a major body posture in which at least one knee touches the ground and the root height is greater than half of the femur length, which is denoted as δ in (9). If only one knee fulfils these criteria, then kneeling is called asymmetric; otherwise, it is symmetric kneeling.

$$((y^t_{lknee} \approx y_0) \vee (y^t_{rknee} \approx y_0)) \wedge ((y^t_{root} - y_0) > \delta) \quad (9)$$

Given that the ground height can be greater than zero (stairs case), we denoted the ground height as y_0.

4.1.7 Squatting

'Squatting' is a major human body posture in which at least one foot touches the ground but not the knee, and the vertical distance between the corresponding hip and foot is smaller than half of the femur length (10). Additionally, the torso must not be horizontal.

$$((y^t_{lfoot} \approx y_0) \wedge (y^t_{lhip} < \delta) \wedge (y^t_{lknee} > y_0)) \vee ((y^t_{rfoot} \approx y_0) \wedge (y^t_{rhip} < \delta) \wedge (y^t_{rknee} > y_0)) \quad (10)$$

Squatting is symmetric when both the knees are bent; it is asymmetric when only one knee is bent.

4.1.8 Lying

'Lying' is a major posture in which the body is in a horizontal or resting position supported along its length. In the proposed approach, this definition is restricted to the torso, i.e. the torso should have an inclination greater than a threshold γ: $\angle(v_{root,cheast}, OY) > \gamma$ (11), which we set at 70° in the conducted experiments. If at least one hip lies on the floor, then the pose is classified as lying on the ground (12), otherwise on an object.

$$(y^t_{lhip} \approx y_0) \vee (y^t_{rhip} \approx y_0) \quad (12)$$

If the two hip joints have approximately the same height (13) and the normal of the frontal plane points down (14), then the pose is lying on the belly. If the mentioned normal points up (15) and the two hip joints have approximately the same height, then the pose is called lying on the back.

$$|y^t_{lhip} - y^t_{rhip}| < \|v_{rhip,lhip}\|/2 \quad (13)$$

$$\angle(f^t, OY) \approx 180° \quad (14)$$

$$\angle(f^t, OY) \approx 0° \quad (15)$$

If the difference between the heights of both the hips is greater than half of the distance between the two hip joints (16), then the pose is lying sideways.

$$|y^t_{lhip} - y^t_{rhip}| \geq \|v_{rhip,lhip}\|/2 \quad (16)$$

4.1.9 Four-Supported

In this rare major posture, the hands and the feet contact the ground but not the root $(y^t_{lfoot} \approx y_0) \wedge (y^t_{rfoot} \approx y_0) \wedge (y^t_{lhand} \approx y_0) \wedge (y^t_{rhand} \approx y_0) \wedge (y^t_{root} > y_0)$. If the belly faces the ground (14), then the position is called 'forward four-supported', or else the back faces the ground (15) and the position is called 'backward four-supported'. Another variant of this posture is when at least one upper limb and one lower limb contact the ground at the same time (17). This variant allows more movements to be performed than the first variant.

$$\left((y^t_{lfoot} \approx y_0) \vee (y^t_{rfoot} \approx y_0)\right) \wedge$$
$$\left((y^t_{lhand} \approx y_0) \vee (y^t_{rhand} \approx y_0)\right) \quad (17)$$

4.1.10 Transition

The transitions between the above-mentioned main postures of the human body are considered here. If the pose cannot be classified as one of the above-mentioned major or minor human body postures, then it is considered a transition posture. The previous and next major postures determine the name of the transition, i.e. the classification of a transitional posture is dependent on the two surrounding main postures. For example, the pose that corresponds to the transitional phase between 'sitting' and 'standing' will be classified as 'standing up'.

5. MULTIPHASE REPRESENTATION OF MOTION

Any human activity can be generally divided into a sequence of simple motions called 'phases'. This division makes the action classification easier and more robust. In the kinesiological analysis of human motion, one tries to divide the considered activity into three phases: preparatory phase, power phase, and follow-through phase [Ham09a], or preparation phase, action phase, and recovery phase [Bar07a]. Here each phase can be further divided into sub-phases so that each sub-phase consists only of some basic joint movements in the directions introduced in Section 4. We use, however, a certain definition of the motion phase and do not distinguish between power phase and other phases. We define the motion phase as a sequence of poses with a common set of features defined above in Section 4. Table 2 summarizes the feature set and the range of values of each feature, where the feature value 'undefined' denotes that this feature is not important in the considered phase, i.e. it can be ignored.

Feature	Values	
Motion Existen	static, dynamic, undefined	
Motion Space	locomotive, non-locomotive, undefined	
Major Posture	standing, sitting, kneeling, squatting, lying, four-supported, transition, undefined	
Minor Posture	standing	upright, bent, leaned, undefined
	sitting	on object, on floor, undefined
	kneeling	symmetric, asymmetric, undefined
	squatting	symmetric, asymmetric, undefined
	lying	{on belly, on back, sideway, undefined } ×{on object, on floor, undefined }
	four-supported	backwards, forwards, undefined
	undefined	
Frontal Motion	{forwards, backwards, fixed, undefined} ∪ $M \times S$	
Vertical Motion	{up, down, fixed, undefined} ∪ $M \times S$	
Sagittal Motion	{left, right, fixed, undefined} ∪ $M \times S$	

Table 2: Summary of introduced features and their possible values, where × stands for the Cartesian product operation, M = {short, mean, long, undefined} and S = {slow, normal, fast, undefined}

This definition of the wide range of high-level features allows the description of the most common human activities in a high language, enabling a comfortable retrieval system. Often, an action that consists of several phases can only be performed starting from a certain phase. In these cases the motion description involves the order of phases. On the other side there are some actions that can be started in more the one phase, such as the kicking action, which consists of three phases and can be started in the first or second phase, where in the first phase the used leg moves backwards to give the strike more power, then it moves forward long fast in the second phase and then moves backwards down to the rest position in the last phase. Here the first phase is optional because kicking can be performed without this phase. Table 3 shows the detailed definition of kicking using the right leg without the optional phase.

Feature	Phase 1	Phase 2
Motion Existence	dynamic	dynamic
Motion Space	non-locomotive	non-locomotive
Main Posture	standing	standing
Minor Posture	undefined	undefined
root Frontal-Vertical-Sagittal Motion	fixed-fixed-fixed	fixed-fixed-fixed
torso Frontal-Vertical-Sagittal Motion	undefined-undefined-undefined	undefined-undefined-undefined
lfoot Frontal-Vertical-Sagittal Motion	fixed-fixed-fixed	fixed-fixed-fixed
rfoot Frontal-Vertical-Sagittal Motion	forward long fast-up mean fast-fixed	backward long fast-down mean fast-fixed
lhand Frontal-Vertical-Sagittal Motion	undefined-undefined-undefined	undefined-undefined-undefined
rhand Frontal-Vertical-Sagittal Motion	undefined-undefined-undefined	undefined-undefined-undefined

Table 3: modeling the motion class "KickR" using the proposed motion model.

Another relative complex example is the jumping action. Jumping can be divided into four phases. In the first phase the feet stay fixed while the root moves down. In the second phase the whole body moves up and forwards, while it goes on forward in the third phase but down. In the last phase the feet are fixed while the root moves up and forwards.

6. ACTION RECOGNITION

Actions to be recognized should be manually modelled and saved in a model database using the developed framework. For each action in the action model database, a finite state machine FSM is created automatically (Fig. 2).

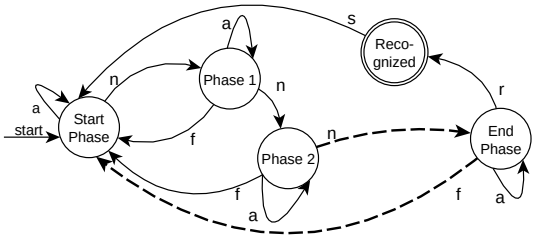


Figure 2: Finite state machine representing the recognition process of a defined action, where 'n' means that the next phase is matched and the current phase can be ended; 'a' means the current phase is matched; 'f' implies failed to match either the current phase or the next one; 'r' means the end phase ended successfully and the motion is recognized; 's' means return to the start phase and start again.

Suppose that an action model consists of n phases $S_1, S_2, \dots, S_n$, where $S_1$is the start phase and S_n is the end phase, then the corresponding FSM is defined as following: $A = (\Sigma, S, s_0, \delta, F)$, where Σ is the input alphabet and consists of all possible pose states; $S = \{S_1', S_1', \dots, S_n'\}$ is the states set and it consists of the action phases whereby each phase is extended to have the following attributes: (1) start time τ, (2) end time σ and (3) an activation flag. s_0 is the initial phase. δ is the transition function and it will be defined later in Fig. 3. F is the set of final states and it consists here of the extended end phase. The input data in each frame consists of the global positions of the used joints as well as the global body orientation. The motion features are computed using this information and then the FSM for each action is updated using the computed current pose state as shown in Fig. 2 and Fig. 3. At the beginning all created FSMs are considered to be in their initial phase. When a new pose is available, the pose state is computed and given to each FSM to update its status as following: if the pose state is compatible with the current FSM phase i.e. the phase is matched, then the phase is retained and the related action is considered active. Otherwise, if the current phase is not matched and it was active in the previous frame, then the phase is considered to be achieved and can be ended if the accumulated motion magnitude and motion speed of each required phase feature are within the desired range and, in this case, the FSM is aggregated to the next phase. Otherwise, the action is cancelled and the FSM is returned to its start phase. If it is assumed that S_t is the pose state of the pose *t*, i.e. S_t is a complete set of the defined features or a complete path in the taxonomy tree, and S_φ is the feature set of the current phase S_φ' of the FSM for the action $\mathcal{M}$, then the global recognition algorithm of the action $\mathcal{M}$ at the time *t* can be stated as follows:

```
1  if S_ω ⊆ S_t then
2      if the current action phase S'_φ is active then
3          set end time of S'_ω  σ = t.
4      else
5          set start time of S'_ω  τ = t.
6          raise the activation flag of S'_φ, i.e. make S'_φ active.
7  else if S'_ω is active and can be ended then
8      if the S'_φ is the end phase then
9          action M is recognized.
10         return to the first phase and reset the activation flag of all phases.
11     else move to the next phase.
12 else return to the first phase and reset the activation flag of all phases.
```

Figure 3: Transition function of the action FSM.

The proposed approach can provide information about the ongoing activity before it is completed, which is an important issue for some application areas such as human–robot interaction, because it enables the robot to response quickly and at the right time.

7. EXPERIMENTAL RESULTS

We developed a framework for action design and action classification from different motion capture databases, namely CMU [Cmu14a], HDM05 [Mue07a], and locally captured data (at our institute). The used data contains distorted walking data. Using our framework we modelled 25 actions manually as explained in section 5. The motion clips were first manually segmented and annotated by two different persons, and then processed by our system. Table 4 shows the actions used in our experiments and the measured evaluation values, where the global precision is about 96.2% and the global recall is more than 98.1%. To begin with, we measured the precision of action recognition as follows: *precision = count of correctly recognized action / count of all recognized actions*. Another evaluation value is the *recall*, which is *the percentage of the count of correctly recognized actions compared to the count of ground truth actions*. An action is considered correctly recognized if the temporal overlap between it and a manually segmented action of the same type is bigger than half the length of the manual action. We measured also the segmentation error as follows: the segmentation error is zero if the difference between the automatic detected cut and the manually created cut smaller than ten, otherwise the segmentation error is equal to this difference minus ten, where a manual created cut is the mean of all manual created cuts (in our case two) of the considered action. The proposed method is able to recognize some particular information about the action such as the marching foot while walking and running, the used hand while punching, or the leg while kicking. All occurrences of most of the defined actions are recognized successfully. An exception is the activity of walking. This is because sometimes the first and last strides of running are recognized as walking. The method failed to match the second phase in the running motion if the feet are not far enough from the ground. This is, however, a minor drawback, because walking and running are similar motions especially in terms of the first and last running strides.

Action Class	Prec-ision	Re-call	Segmentat-ion Error
WalkL	0.94	0.99	2
WalkR	0.93	0.99	1
RunL	0.98	0.94	0
RunR	1	0.96	0
BoxL	1	0.96	11
BoxR	0.96	1	19
KickR	1	1	10
KneeKickR	1	1	27
SideKickR	1	1	23
Jump	1	1	11
JumpJacks	1	1	7
StandUp	1	1	55
SitDown	1	1	14
Hop2Legs	1	1	71
HopR	1	1	31
HopL	1	1	20
SwingArmsSagittal	1	1	11
SwingArmsTravers	1	1	26
SwingArmsCircular	1	0.94	14
ChoppingL	1	1	4
ChoppingR	1	1	19
Fight	1	1	28
DrinkR	1	1	18
Throw	1	1	57
Squat	1	1	32

Table 4: Results of the experiments, where 'L' stands for left and 'R' for right and it refers to the active limb during the action.

The classification speed is linear with the number of actions to be recognized. The mean recognition speed for a model database of 25 actions amounted ~1200 fps on a computer running Windows 8 with AMD A4-4300M APU processor, 2.50GHz and 4.00GB RAM. If the database were hypothetically extended to contain 250 actions, then the speed would sink to ~120 fps. This means that our method can scale to large model databases and can still perform well in real time.

Compared to some other works which were evaluated using data from the same data sources which we used ,namely the HDM05 and CMU, the proposed method produces better results as shown Table 5. However this comparison might be unfair because the used datasets might be slightly different and the classes and numbers of considered actions are also different.

Action Class	[Cho13a]	[Lei14a]	[Zha12a]	Proposed
All	0.95	0.9492	0.927	0.962
Walk	-	~0.975	0.923	0.935
Run	-	~0.975	0.989	0.99
Hop	-	~0.95	1	1
Box	-	~0.86	-	0.98
Squat	-	~0.94	-	1

Table 5: Precision of some other works where „-“ stands for unknown accuracies and „~“ stands for those read from a diagram picture.

8. CONCLUSION AND FUTURE WORK

In this paper a set of high-level semantic features are introduced and employed in a multiphase motion representation that enables an efficient recognition and retrieval of motion capture data with high accuracy. The introduced features as well as the multiphase representation of motion are inspired by kinesiology, and hence the proposed method mimics the human mind by motion perceiving and analysing what enables it to perform very well. It can also work online and offline in real time. The recognizable motion database can be extended easily and in a short time, because our method does not require any training time. The experiments made on large databases from different sources, as well as on distorted data, proved that the proposed method scales well to other data sources. As future work we plan to extend this method so that it can also classify single poses, static clips, and static gestures.

9. ACKNOWLEDGEMENTS

Some of the datasets used in this work were obtained from mocap.cs.cmu.edu while some other sets were obtained from HDM05.

10. REFERENCES

[Bar04a] Barbic, J., Safonova, A., Pan, J. Y., Faloutsos, C., Hodgins, J. K. and Pollard, N. S.. Segmenting motion capture data into distinct behaviours. Graphics Interface, 185-194 , 2004.

[Bar07a] Bartlett, R.. Introduction to Spotrs Biomechanics: Analysing Human Movement Patterns 2nd Edition. ISBN 0-203-46202-5, Routledge, UK, USA and Canada, 2007.

[Bau14a] Baumann, J., Wessel, R., Krüger, B. and Weber, A.. Action Graph: A Versatile Data Structure for Action Recognition. International Conference on Computer Graphics Theory and Applications, 2014.

[Cho13a] Cho, K. and Chen, X.. Classifying and Visualizing Motion Capture Sequences using Deep Neural Networks. arXiv preprint arXiv:1306.3874, 2013.

[Cmu14a] CMU Graphics Lab Motion Capture Database, http://mocap.cs.cmu.edu/search.php?subjectnumber=86. Date of Access March, 16, 2016.

[Cop15a] Coppola, C., Martinez Mozos, O. and Bellotto, N.. Applying a 3D qualitative trajectory calculus to human action recognition using depth cameras. IEEE/RSJ IROS Workshop on Assistance and Service Robotics in a Human Environment, 2015.

[Gre05a] Greene, D. P. and Roberts, S. L.. Kinesiology: Movement in the Context of Activity 2nd Edition. ISBN 0-323-02822-5, Elsevier Inc., USA, 2005.

[Ham02a] Hamilton, N. and Luttgens, K.. Kinesiology: Scientific Basis of Human Motion 10th Edition. ISBN 0-07-112243-5. McGraw-Hill, USA, 2002.

[Ham09a] Hamill, J. and Knutzen, K. M.. Biomechanical Basis of Human Movement 3d Edition. ISBN-13: 978-0781791281 ISBN-10: 0781791286, Lippincott Williams & Wilkins, USA, 2009.

[Jin07a] Jin, Y. and Prabhakaran, B.. Semantic Quantization of 3D Human Motion Capture Data Through Spatial-Temporal Feature Extraction. MMM 2008, LNCS 4903, pp. 318–328, 2007.

[Lei14a] Leightley, D., Li, B., McPhee J., Hoon Yap, M., Darby, J.. Exemplar-Based Human Action Recognition with Template Matching from a Stream of Motion Capture. 11th International Conference, ICIAR 2014, Vilamoura, Portugal, 2014.

[Liu06a] Liu, G., Zhang, J., Wang, W. and McMillan, L.. Human Motion Estimation from a Reduced Marker Set. Proceedings of the 2006 symposium on Interactive 3D graphics and games, ACM, USA, 2006.

[Mue07a] Müller, M., Röder, T., Clausen M., Eberhardt, B., Krüger, B. and Weber, A.. Documentation Mocap Database HDM05 (Part-Scene #1). Germany, 2007.

[Sal15a] Salamah, S., Zhang, L., Brunnett, G.. Hierarchical Method for Segmentation by Classification of Motion Capture Data. Virtual Realities pages 169-186, ISBN 978-3-319-17043-5, 2015.

[Zha11a] Zhang, L., Brunnett, G. and Rusdorf, S.. Real-time Human Motion Capture with Simple Marker Sets and Monocular Video. Journal of Virtual Reality and Broadcasting, Volume 8, no. 1, 2011.

[Zha12a] Zhang, M. and Sawchuk, A. A.. Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach. Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, USA, 2012.

Interactive Visual Analysis of Multidimensional Geometric Data

Igal Milman
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
115409, Moscow, Russia
igalush@gmail.com

Victor V. Pilyugin
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
115409, Moscow, Russia
VVPilyugin@mephi.ru

ABSTRACT

One of the most important tasks in modern world is to find solutions to problems of processing and analyzing multidimensional data. In this paper we present an approach for cluster analysis of multidimensional geometric data. Some definitions and extensions of classical cluster analysis problem is given. Our approach is based on the visualization method. Suggested approach allows us to analyze multidimensional data and distances in multidimensional Euclidean space using three-dimensional spatial scenes and shows an easy way for cluster analysis and anomaly discovery. An example of solving the problem of analysis of financial multidimensional data of credit organizations is also presented.

Keywords

Visual analysis, cluster analysis, multidimensional data analysis, visualization.

1. INTRODUCTION

One of the most important tasks in modern world is to find solutions to problems of processing and analyzing multidimensional data. Different methods and procedures, both automatic and interactive, have been developed to solve such problems. Visual methods take a special place among the data analysis problem solving methods.

However, a careful study of publication focused on the description of specific applications that use visual methods, allows us to state that in reality, interactive multidimensional data analysis systems often have a lower value than systems displaying results gotten using data analysis methods. As an example we can use situational alerts system AdAware [1], system of visual analysis system that is used to solve problems in aircraft manufacturing [2], system of visual analysis of text data VxInsight, software package SAS Visual Analytics [3], created to process and analyze large volumes of financial and economic data. All above mentioned systems are industrial and commercial products; they provide users with a great number of interfaces and data visualization capabilities. However, while all of these systems, in fact, are set to process multidimensional data internally and present results in the form that is convenient for the user, they don't give him an option to work directly with data clouds using multivariate visual display of the data.

As practice shows methods of parallel coordinates [4], Chernoff faces [5], Andrews plots [6] and other mnemonic graphic images are widely used for such visual representation for multidimensional data. Such images have a set of settings corresponding to the coordinates of multidimensional point. And, by comparing that images, one can cluster the initial data. For more info about such methods see works [4-6]. These methods do not allow the user to use any kind of metrics to understand the difference between objects. That is a crucial point, if the analyst has to answer the question "why does this objects are similar?"

This article discusses an original algorithm that we developed to solve problems of multi-dimensional geometric data analysis. Justified choice of the method used preceded by the development of the algorithm and based on the algorithm we created an interactive software package for visual analysis of multidimensional data. This method and algorithm are different from others since they provide the user with the ability to work directly with the original multidimensional data - there is no initial numeric processing of the original multidimensional data, and that allows analyst to manipulate directly with input data and visually analyze the results.

2. STATEMENTS OF THE GEOMETRIC DATA ANALYSIS PROBLEM

In this article, a geometric data refers to a set of points $(x_1, x_2, \dots, x_n)$ of Euclidean space E_n with predetermined metric tensor $\rho(x, y)$, which may be prepared by geometrization of any domain data. The task of geometry data analysis is understood as a problem of extended cluster analysis, as well as the imposition of additional statements on the mutual

positions of the points in multidimensional geometrical space.

2.1. The classical problem of cluster analysis

The problem of cluster analysis is one of the classical problems of data analysis. Setup of goals of cluster analysis includes the following:

Given: set of points $G = \{x_1, x_2 \dots x_m\}$, where $x_i = (x_i^1, x_i^2, \dots, x_i^n)$

Required: divide subset G_i from G, in a way that:

1) $G_i \cap G_j = \emptyset, \forall I, j, i \neq j$
2) $\cup G_i = G$

In the classical statement of the cluster analysis problem, subset G_i is called *cluster* and must satisfy the following conditions (with certain function for calculating the distance $\rho(x, y)$ and maximum intra-cluster distance d):

1) $\forall x, y \in G_i, \rho(x, y) \leq d$
2) $\forall x \in G_i, \forall y \in G, y \notin G_i, \rho(x, y) > d$W

2.2. Extended problem of cluster analysis

Depending on the distance $\rho(x, y)$ between the points and the parameter d, that may be changed during the analysis process, it is possible to visually distinguish the following subset of multidimensional points:

1. *Cluster* — classical cluster.
2. *Remote (anomalous) point* — a point x_i is remote, if $\forall y \in G, \rho(x_i, y) > d$. We may say, that distant point — is a cluster of the size of one. However, these points may be of particular interest for the analyst.
3. *Bunch* — a subset of points with most distances between points not exceeding the preset d value.
4. *Quasi-remote point* — a point that is not remote, but at the same time is not included in a bunch or a cluster at the given grouping.

Note, that the analyst selects bunches and quasi-remote points during the process of solving the above-mentioned problem of the analysis. These concepts are useful for the analyst in the process of solving the problem of geometric data analysis.

Allocation of bunches and quasi-remote points allows the analyst to focus on these objects during the process of changing the parameter d. Therefore, if there is an allocated quasi-remote point, it is necessary to gradually change the value of d, to find out the conditions under which a point would become anomalous. Similarly, when allocating bunches, it is necessary to change d to try to obtain a cluster.

2.3. The statements of the relative positions

In the process of solving the problem, statements of the following types are made:

- Point x_i belongs to subset G_j when $d = d_k$
- Subset G_j is a cluster
- Subset G_j is a bunch
- Point x_i is an anomalous point
- Point x_i is a quasi-remote point.

As a result, in this publication we are solving the problem of partitioning of the original multidimensional geometric data into subsets, such as clusters and remote points, as well as the allocation of bunches and quasi-remote points supporting the problem solving process when the analyst changes maximum intra-cluster distance d.

3. THE PROPOSED METHOD

To solve this problem, it is proposed to use the visualization method. Theoretical aspects of the solution for data analysis problems with this method using the scientific data as an example are given in [7]. The essence of the visualization method is to divide the original problem into two consecutively solved sub problems. First problem, solved by a computer, is to obtain a representation of the analyzed data in a graphical display (the problem of data visualization). Second one, is to analyze the graphic image and interpret the results of the analysis against the original data. This problem is solved directly by man.

It is emphasized, that in this method the visual analysis of a graphical representation of the analyzed data is to qualitative analysis of the spatial scene that within this method is corresponding to the analyzed data. I.e. used graphics are means to naturally and comfortably for the analyst to visually analyze the spatial scene, followed by the interpretation of the results correlated to the original data. An algorithm for solving the first problem involves the following steps:

1. Sourcing — receiving original data for visualization pipeline.
2. Filtering — pre-processing the original data. During that step an interpolation of missing data, data decimation and data smoothing can be applied. In general, this step may be absent.
3. Mapping — on this step, the filtered data is mapped to spatial scene. This step is one of the most important and time-consuming in the first task.
4. Rendering — obtaining the resulting graphics of spatial scenes.

The second objective is to analyze the resulting graphics that is visual analysis of spatial scenes. This step cannot be strictly formalized, its effectiveness depends on the experience of the person performing

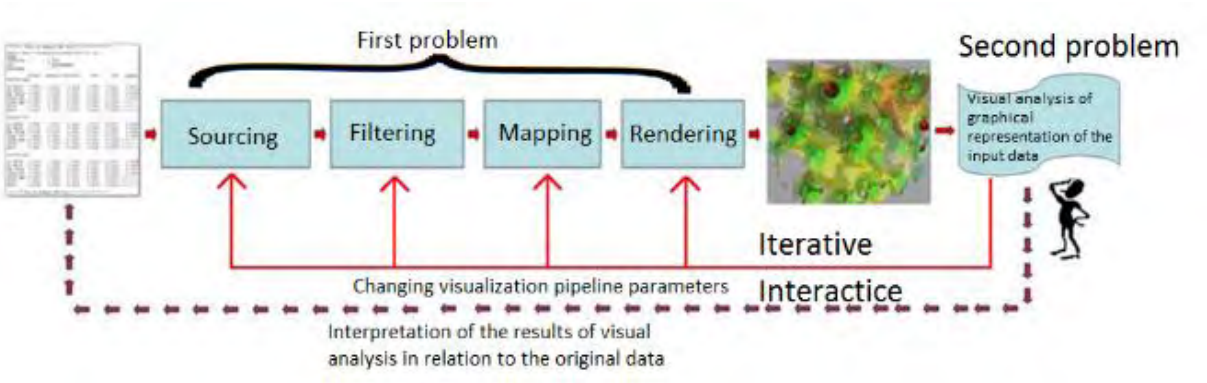


Figure 1. The general scheme of the visualization method

the visual analysis and his tendency for spatially-shaped thinking. Looking at the resulting image, a person can solve 3 main objectives: analysis of the shape of spatial objects, analysis of their mutual disposition and analysis of graphic attributes of spatial objects. The results of the solutions of these three problems, as indicated above, are interpreted with respect to the original data.

In the analysis of the graphics, the user can either be satisfied with the conclusions, or it may come back to one of the stages of the first task to set other values of the visualization pipeline parameters. Therefore, the process of solving the analysis' problem using the visualization method is iterative and interactive. The general scheme of analysis' problem solving is shown in figure 1.

3.1. Description of the basic idea of the algorithm

Under this method, we proposed an original algorithm for solving the problem of visualization. The basic idea is that in the original n-dimensional space E_n an additional construction is undertaken. If the distance between two n-dimensional points x_i, x_j is not more than pre-assigned d ($\rho(x_i, x_j) \leq d$), line segment is drawn between two points in the original space. Accordingly, the original analyzed data appeared to be m multidimensional points and some multidimensional line segments (depending on d). Then projection of the original space on the selected by the analyst 3-dimensional space X_i, X_j, X_k is performed. Next, a spatial scene is constructed using the following rules:

- points correspond to spheres with preassigned radius;
- line segments correspond to cylinders with preassigned radius.

The color of the spheres is set to be the same, and the color of the cylinders depends on the distance in the original space. The smaller the distance, the redder the cylinder between the spheres. When setting up the color of the cylinder in the RGB palette, the color will be set as follows:

$$RGB = [255, 0, 0] + [-255, 150, 255] * \frac{\rho(x, y)}{d}$$

Setting various colors to cylinders allows making statements about the distance in the original n-dimensional space while visual analysis is performed in the 3 dimensional spatial scene.

In case of several n-dimensional points are projected into one 3-dimensional point, we should move for a bit one of the 3-dimensional point in a such way, that the points are not overlaying anymore. Due to the fact that described algorithm assumes analysis of the distance between n-dimensional points, such transformation does not violate the process of visual analysis of the spatial scene and the analysis of the initial data as a whole. So, even in that case, that algorithm is valid.

3.2. Detailed description of the algorithm

The algorithm of solving the geometric data analysis problem is represented by the following steps:

1. Input of initial data.
2. Choosing the distance formula.
3. Setting the maximum intra-cluster distance d, calculating distance between every couple of points in the original n-dimensional space.
4. Entering visualization parameters (radius of the spheres and cylinders, space X_i, X_j, X_k for projection).
5. Projecting objects of the original n-dimensional space into chosen in step 4 3-dimensional space.
6. Creating of spatial scene.
7. Visualization and analysis of spatial scene.
8. If not all the necessary information is obtained, it is necessary to go back to step 3.
9. The results of the analysis were then interpreted relative to the original multidimensional geometric data.

Therefore, the algorithm of solving the problem is an interactive and iterative.

Now we are estimating the complexity of one cycle of the algorithm (steps 3-8). If the number of points

1776	0.518847319	1.532329371	3.609459961	3.01106337	3.04738502	5.72919772	1.973569238	73.61787413	9.019037609
1792	1.39669798	0	0	0.187793432	0.348829902	4.034758471	23.51642663	79.79540476	14.41660886
1810	0.487264847	1.273238018	3.732998385	0.810198473	1.836966841	0.754432152	3.075156652	81.06465639	2.575766957
1942	1.55485847	3.456152257	6.099577778	1.065347058	1.288593266	1.660005231	0.091947455	76.48378174	12.01384122
1961	0.117578451	0.850914851	3.123172123	0.891393673	0.235722398	1.993900825	0.242739354	81.35683766	2.402768244
1971	2.821605938	2.169194629	7.277215747	3.197183488	1.299417418	2.189164803	2.399673705	74.65238431	0
1978	0.97372923	2.252979064	6.501586157	3.799072284	1.491681846	1.152002579	3.160030324	80.17056587	17.50707353
2119	0.183484124	0.231239772	0.717639287	0.055330121	0.315534217	0.024389973	2.564332931	78.13577277	0.30034603

Figure 2. Fragment of the original data

is defined as n, then the number of cylinders will be $n * \frac{n-1}{2}$. Thus,

$$T(n) = O\left(n * \frac{n-1}{2}\right) = O(n^2)$$

$$M(n) = O\left(n * \frac{n-1}{2} + n\right) = O(n^2)$$

where $T(n)$ shows the dependence of the operating time on the volume of the input data (n);
$M(n)$ shows the dependence of the consumed memory vs. the volume of input data (n).

3.3. Implementation in Maxscript and C ++ (VTK)

This algorithm has been implemented in two different ways. First it was implemented using the programming language maxscript (3ds Max environment) due to its simplicity and richness of conceptual apparatus and therefore high speed programming on it. Because of the high complexity of the resulting spatial scene and a large number of objects on it (with a 90-points in the scene is displayed up to 8000 objects), as well as high frequency of the redrawing of the spatial scene, rendering takes a long time (4-5 minutes), and a greater number of original points causes a memory overflow. An additional constraint imposed by the 3ds Max is the complexity to design the user interface and simplicity of the tools for its implementation.

Then, given these drawbacks of 3ds Max usage, it was decided to move to the C++ programming language using VTK 7.0 library for visualization and programming environment Visual Studio 2013. The program uses a total of 13 user-defined classes and 5 user-defined types.

Optimization of calculations, the usage of a compiled language instead of interpreted and simpler visualization software in the C++ version of the software helps streamline the rendering process. When we process data contained of 81 points, using the software, instead of 4-5 minutes before, it took a few seconds now. The amount of RAM required for such data in 3ds Max was close to 1GB, while the software written in C ++ requires only 70MB. As a result, the transition to a new language will allow to analyze much larger volumes of data, as well as to create user friendly interfaces and tools for manipulation of spatial scenes.

4. EXAMPLE OF USING THE SOFTWARE

The described software tool has been tried to solve the problem of data analysis on the activities of credit organizations, presented in tabular form. [8]

4.1. Characteristics of the original data

Original data is multidimensional tabular data obtained from the financial statements of 81 credit organizations with 9 parameters for the second half of 2013 and the first half of 2014. A separate table was created for each month.

The tables have been created as follows: rows contain information about credit organizations, and columns contain parameters of those organizations. A total of 13 months was considered, so there are 13 tables. A fragment of the original data is shown in figure 2.

We tried to solve the problem of analysis of similarity of credit organizations. The goal was to highlight anomalous objects at different values of the similarity measures.

It was necessary to allocate credit organization diverged from other ones.

To solve the above problem using the proposed method we perform a geometrization of the problem. Geometrization allows us to transfer initial data from any domain to geometric data. Thus, after the geometrization, we can use described algorithm for any kind of data.

Geometrization will be performed as follows:

1. Each credit organization (each row of the table) will be assigned to a point of 9-dimensional Euclidean space.
2. credit organizations parameters (columns) will be interpreted as coordinates of points in the 9-dimensional space.
3. The distance in Euclidean space will be interpreted as a measure of the difference between credit organization. In this problem we use the Euclid distance:

$$\rho(x, y) = \sqrt[2]{\sum_{i=1}^{9} (x_i - y_i)^2}$$

4.2. Analysis

The algorithm of usage of the software requires to set a large value of the maximum intra-cluster distance.

In other words, select a value d, in which all spheres are connected by the cylinders.

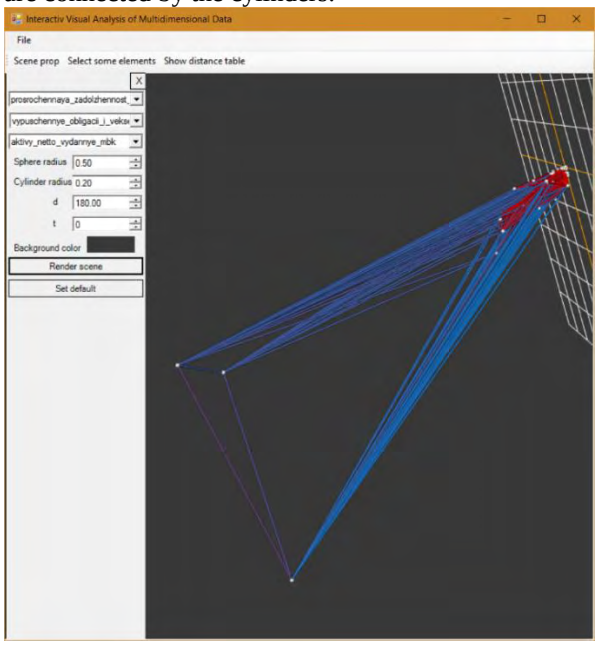


Figure 3. A graphical projection of the space scene, if *d=180*.

Figure 3 demonstrates a graphical projection image of the space scene, if *d=180*. As it is seen, all spheres are linked to each other and, therefore, respective multidimensional points made up a cluster. This value of *d* will be used as the initial value and it has to be reduced later on.

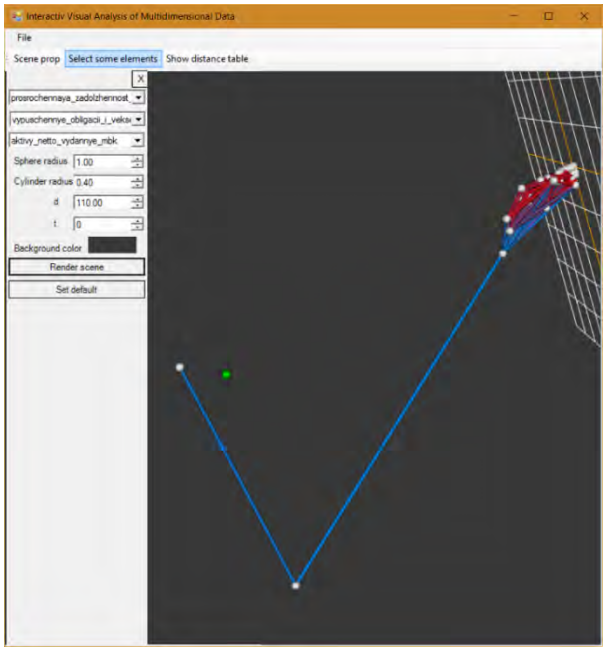


Figure 4. A graphical projection of the space scene, if d=110

Figure 4 illustrates a graphical projection image of the space scene, if d=110. We marked the sphere that has no connections with at that value of *d* as well as the appropriate remote point (ID=2748) by green color. Later on the color of the sphere will define the point in the multidimensional space fixed by the analyst, i.e. a predetermined color will allow us to trace any given point. With a further decrease of the d, highlighted green point will not change its properties, and there is no further need for its consideration. A bunch, containing of two white spheres, can be highlighted with this value of parameter d. Perhaps with further decreasing of the d, it will be turned into a cluster or two remote points.

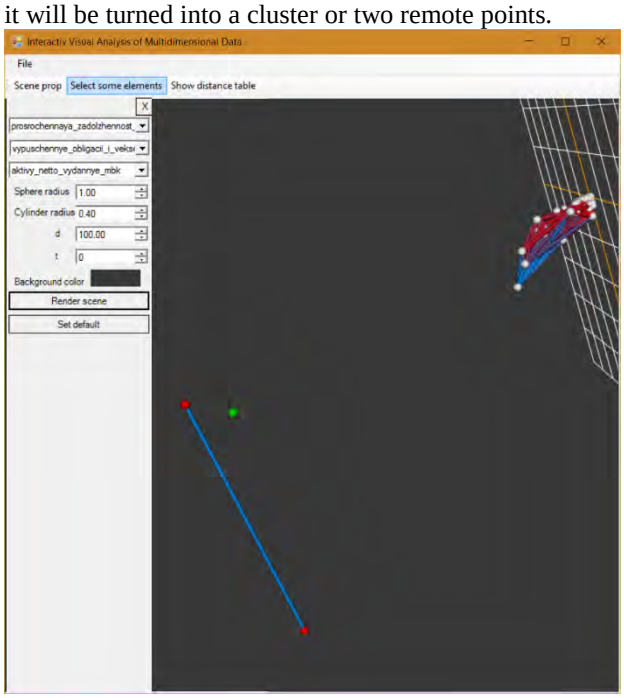


Figure 5. A graphical projection of the space scene, if d=100

Figure 5 represents a projection image of the space scene, if *d=100*. One can see that two spheres have been disconnected from others and two corresponding multidimensional points (ID=354 and 1000) formed a cluster. Based on the color of the cylinder being close to bright blue, the distance between these points is close to d. We will color these spheres (and the corresponding multidimensional points) in red.

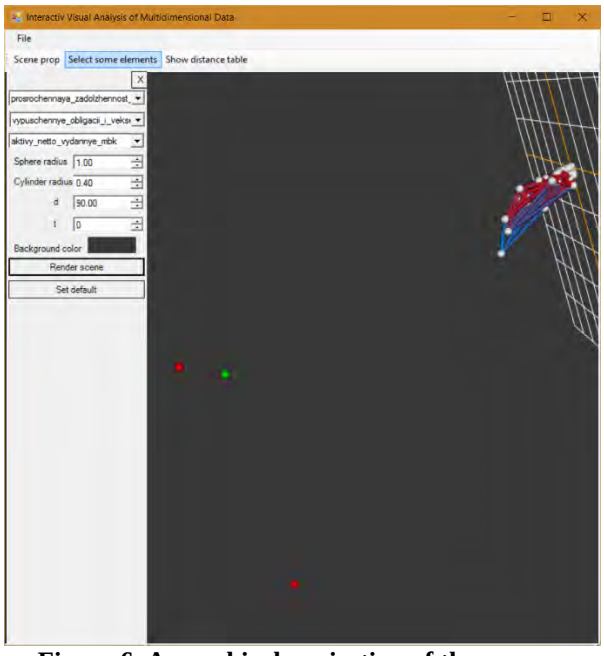


Figure 6. A graphical projection of the space scene, if *d=90*

Figure 6 demonstrates a graphical projection image of the space scene, if *d=90*. The cylindrical linkage between them is gone, which means the distance between the corresponding points greater than 90. In this case, the points turned into remote points. These points will not affect the further analysis.

Therefore, an analyst can choose green and two red points as desired remote points or he can continue the analysis by implementing further decrease of d and finding new remote points according to his knowledge of the specifics of the analysis of credit organization problem being solved [8]. In other words, analyst can conclude that there are three remote points. Moreover, in the process of solving the problem, with a sequential decreasing of the d, analyst may form the following additional conclusions:

1. When $d \leq 110$, point with ID=2748 is anomalous.
2. When $90 < d \leq 100$, points with ID=354 and ID=1000 form cluster of size two.
3. When $d \leq 90$, point with ID=354 is anomalous.
4. When $d \leq 90$, point with ID=1000 is anomalous.

5. CONCLUSION

In this paper we described the original algorithm for solving the problem of the analysis of multidimensional geometric data. This algorithm, in disparity to other algorithms that are using the visualization method, offers the user the ability to work directly with the original multidimensional data using visualized projection of that data in three dimensional space. The original numerical processing of multidimensional source data is not performed; instead, the analyst directly manipulates the source data and then performs visual analysis of the resulting data.

This algorithm was implemented using the programming language C ++. The resulting software tool has been tested on the data on the activities of credit organizations. As a further development of the system, it is proposed to add a number of tools for viewing spatial scene with different values of the maximum inter-cluster distances.

REFERENCES

[1] Y. Livnat, J. Agutter, S. Moon, and S. Foresti, 2005. *Visual correlation for situational awareness*. In IEEE Symposium on Information Visualization, pp. 95-102.

[2] D.N. Mavris, O.J. Pinon, D. Fullmer Jr, 2010. *Systems design and modeling: A visual analytics approach*. 27th Congress of International Council of the Aeronautical Sciences ICAS.

[3] *SAS the power to know*, [Online]. Available: http://www.sas.com/en_us/home.html. [Accessed 26 1 2016].

[4] A. Inselberg. *Multi-dimensional graphics: algorithms and applications*. EUROGRAPHICS'86, North-Holland (1986) pp. 7-18.

[5] David L. Huff, Vijay Mahajan and William C. Black. *Facial Representation of Multivariate Data*. The Journal of Marketing, Vol. 45, No. 4 (1981), pp. 53–59.

[6] Andrews D.F. *Plots of High-Dimensional Data*. Biometrics. Vol. 28, No. 1, Special Multivariate Issue (Mar., 1972), pp. 125-136

[7] A. Pasko,V. Adzhiev, E. Malikova, V. Pilyugin, 2013. *Some Theoretical Issues of Scientific Visualization as a Method of Data Analysis*. the Lecture Notes in Computer Science series.

[8] I.E. Milman, A.P. Pakhomov, V.V. Pilyugin, E.E. Pisarchik, A.A. Stepanov, Yu.M. Beketnova, A.S. Denisenko, Ya.A. Fomin, 2015. *Data analysis of credit organizations by means of interactive visual analysis of multidimensional data*. Scientific Visualization. Vol. 7. No. 1. Pp. 45 - 64.

[9] J. Thomas and K. Cook, 2005. *Illuminating the Path: Research and Development Agenda for Visual Analytics*. IEEE-Press.

[10] D.A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, 2008. *Visual Analytics: Scope and Challenges*. Lecture Notes in Computer Science, pp. 76-90.

[11] J. J. v. Wijk, 2005. *The value of visualization*. IEEE Visualization. Pp. 79-86.

[12] *What is Visual Analytics?*, [Online]. Available: http://www.visual-analytics.eu/faq/.

[13] D. Keim, G. Andrienko, J.D. Fekete, G. Carsten, J. Kohlhammer, 2008. *Visual Analytics: Definition, Process and Challenges*. Information Visualization - Human-Centered Issues and Perspectives, pp. 154-175.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

A User-centered Approach for Optimizing Information Visualizations

David Baum, Pascal Kovacs, Ulrich Eisenecker and Richard Müller
Leipzig University
Grimmaische Strasse 12
04109 Leipzig, Germany
[baum, kovacs, eisenecker, rmueller]@wifa.uni-leipzig.de

ABSTRACT

The optimization of information visualizations is time consuming and expensive. To reduce this we propose an improvement of existing optimization approaches based on user-centered design, focusing on readability, comprehensibility, and user satisfaction as optimization goals. The changes comprise (1) a separate optimization of user interface and representation, (2) a fully automated evaluation of the representation, and (3) qualitative user studies for simultaneously creating and evaluating interface variants. On the basis of these results we are able to find a local optimum of an information visualization in an efficient way.

Keywords

Evaluation, Information Visualization, Optimization, Usability, User-centered design

1 INTRODUCTION

Over the last years, a considerable number of visualizations has been presented [CZ11, LBAAL09, LCWL14, TC08, vLKS+11, ZSAvL14]. The benefit of a specific visualization depends on many factors, such as addressed stakeholder (e.g. project manager, analyst, scientist, or developer), the chosen methods of representation and interaction, and the supported tasks [LCWL14, vLKS+11]. Because of the number of factors and their connections evaluating visualizations is a big challenge. Nevertheless, in most cases more time is spent on developing entirely new visualizations than to evaluate them and some of them have not been evaluated at all [WLR11, TC08].

Empirical quantitative studies are an established type of evaluation and can prove that one visualization is superior over another one. However, planning, conducting and analyzing such a quantitative study is difficult, time-consuming and causes a huge effort [And06, CC00, KSFN08, LBIP14, Pla04]. Especially, recruiting a sufficient number of participants is hard if they have to meet certain criteria such as specific profession (e.g. software developer with industrial experience). Tasks are another critical aspect in such studies, because simple tasks are easier to create but are not representing a real world scenario, which is a thread to the external validity[vLKS+11]. Complex tasks on the other side are more difficult to create, because of the higher risk of misinterpretation by the participants. Despite of these difficulties, a quantitative study may lead to significant results, but often gives not enough insight into the details, why a visualization is superior [LBIP14]. Furthermore, the choice which visualizations or visualization variants should be investigated is a critical part of what the results are useful for, but in most cases it is not exactly reasoned.

These obstacles apply to evaluation of visualizations in general and even more to their optimization, because to achieve a satisfying visualization several improvements and therefore evaluations have to be done. Thus, not only a single visualization has to be evaluated but also several variants differing in representation details and interaction options. Due to the complexity of most visualizations the amount of possible variants is far too high for evaluating every variant. Therefore, it is necessary to apply reasonable strategies to reduce the number of variants to be evaluated down to a manageable number.

In this paper we present our approach for the optimization of visualizations regarding readability, comprehensibility, and user satisfaction, derived on our experience of evaluating software visualizations. We combine computational, qualitative and quantitative methods into a well-structured and repeatable process, based on existing processes for user-centered design (UCD) and considering the specific characteristics of information visualizations. By adopting this process a researcher can reduce time and effort finding a local opti-

mum in an efficient heuristic way to improve any visualization.

2 APPROACH

User-centered approaches are usually based on at least four iterative steps [PvES06]:

1. identify need for UCD
2. produce design solutions
3. evaluate designs against requirements
4. final design

Compared to existing approaches we alter steps 2 and 3 by optimizing user interface (UI) and representation separately. Thereby, for each part of the optimization the most efficient and suitable method can be chosen. We propose an aesthetics-based approach for representation optimization (section 3) and user studies for UI optimization (section 4) as shown in figure 1. Like existing UCD approaches, the whole process is repeated until a certain criterion is reached [WKD09]. Depending on the motivation of the optimization this can be, e.g., a targeted deadline or the detection of merely non-significant improvements.

Munzner [Mun09] introduced a four-layered model for visualization design and validation. According to this model, our approach refers to the layer *encoding/interaction technique design*.

3 OPTIMIZE REPRESENTATION

Aesthetics are visual properties of a representation that are observable for human readers as well as measurable in an automated way [Bau15]. Some of them affect the readability and comprehensibility significantly, either in a positive or in a negative way [Pur97]. The effects on user performance can be measured in a quantitative study using the time a user needs to solve a task and the number of errors he makes [Hua14]. Based on aesthetics, representations that are optimized for readability and comprehensibility can be designed.

As aesthetics emerge from the properties of the depicted elements, such as color, shape, size, and positioning, they are specific for every representation [BRSG07, PAC02]. For some basic representations like node-link diagrams aesthetics and their influence on readability and comprehensibility are well-understood [BRSG07]. In this case the process of optimization becomes easier, since some part of the work is already done. The gathered knowledge about aesthetics can be reused in further iterations. Hence, the effort is reduced with every iteration and quantitative studies might even become obsolete.

3.1 Produce Representation Variants

The previous work of Baum [Bau15] describes how the repertory grid technique can be used to identify relevant aesthetics for any representation in a structured and reproducible way. The resulting list of aesthetics is narrowed down by two requirements that have to be fulfilled. First, no information may be lost; second, there must be a significant effect on user performance. A solely aesthetics-based optimization of readability is not meaningful if the changes imply an adulteration of the visualized content. For example, a layout algorithm might convey information via the order of the depicted elements. If this order is changed, e.g., to reduce space consumption, the result may be more readable but some information is tampered. Further, it is unlikely that all identified aesthetics have a significant effect on user performance, based on the experiences with node-link diagrams [WPCM02]. To reveal the relations between aesthetics and user performance quantitative studies are still required. Every examined visualized data set is based on the same visualization but holds different values for one or multiple aesthetics. Measuring the time needed by a user to solve a task and the number of errors made while doing so yields two important findings. First, the aesthetics that have a significant effect on user performance; second, the weighting of those aesthetics since they differ in their impact.

Eventually, one or more variants of the original representation can be created with respect to the most influential aesthetics, e.g., by applying another layout algorithm. Except during the first iteration the results of the user studies can be used as additional source of information. Producing variants still requires the creativity of the researcher since aesthetics only determine the goal of the optimization but not how it can be achieved. For example, our approach does not help to develop completely new layout algorithms, but aesthetics provide assessment criteria for automatic evaluation.

3.2 Evaluate Representation Variants

Aesthetics allow a fully automatized evaluation [Pur97]. For every created variant its effect on readability and comprehensibility can be automatically calculated by making use of the gathered information. Hence, the evaluation is very efficient and even a large amount of variants can be evaluated without difficulty. The outcome of the evaluation is a representation variant that will be further optimized.

4 OPTIMIZE INTERACTION

The interaction between the user and a visualization is realized through the UI, which is a complex combination of interaction techniques (ITEC). Yi et al. [YaKSJ07] define ITECs in information visualization as *"[...] the features that provide users with the*

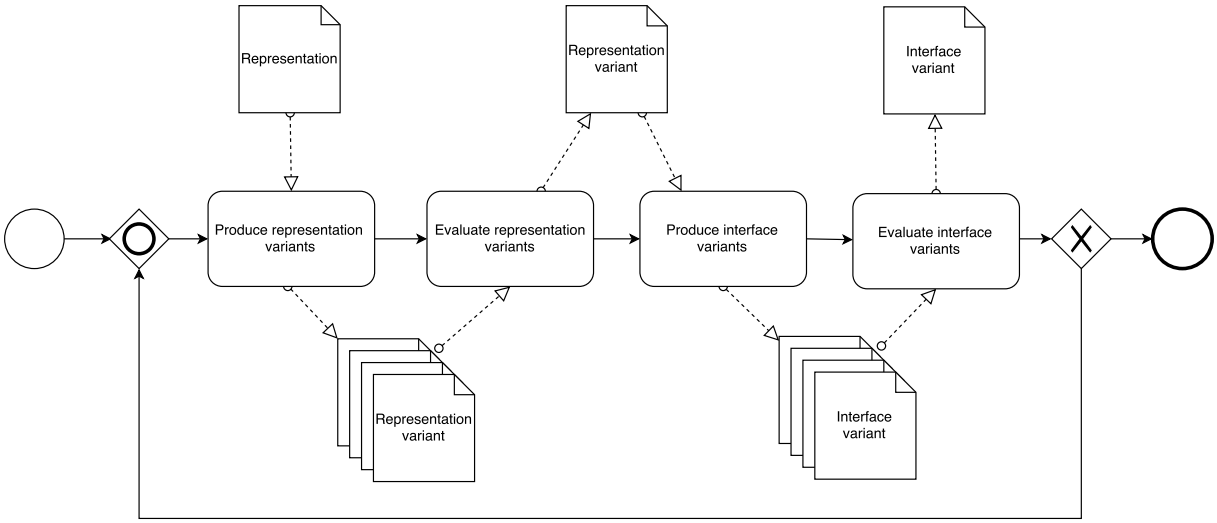


Figure 1: Optimization process for information visualizations

ability to directly or indirectly manipulate and interpret representations". To categorize ITECs they propose a taxonomy of seven categories: *select, explore, reconfigure, encode, abstract/elaborate, filter, and connect*. Hence, evaluating the interaction with a UI via ITECs could be done in four different levels of detail, from low to high, by

- comparing full UIs against each other,
- integrating ITECs in the UI,
- pairwise comparisons of ITECs of one category, and
- scrutinizing details of a single ITEC.

With the target of optimizing the interaction as a whole, a quantitative evaluation in one of these levels is not suitable, because either the reasons why one UI is superior over another UI can not be identified or the context of the target domain is lost when evaluating only the details of one ITEC. A quantitative evaluation of all four levels is also not feasible, because of the huge effort and the difficulties, even when comparing only two variants per level[LM08]. Furthermore, the space of possible variants is huge, thus choosing the variants for further evaluation and improvement is a critical part.

Therefore, we propose iterative qualitative user studies in a within-subject design as a heuristic to find a local optimum in the huge space of possible UI variants. One iteration consists of a couple of runs, where every participant solves a set of randomized tasks using an optimized representation variant and more than one UI variant. Each UI variant differs in at least one detail of ITECs, e.g., one variant has zoom by mouse wheel to the position of the cursor, the other one zooms by double click on an element, and a third one zooms twice as fast as the second one using an addition button. The first iteration starts with some UI variants chosen by the researcher, which are derived from his own ideas or by other visualizations or guidelines. Further iterations may contain subsequent UI variants triggered by analyzing the feedback of participants. Additionally, tasks may be altered, bugs in the visualization can be fixed, and ideas for representational variants could be identified, which will be used during representation optimization. If the optimization process is terminated a final UI is derived from the evaluation of the investigated UI variants.

To get as much detail about the interaction as possible, qualitative data is collected about each UI variant and also about the tasks and their descriptions. Therefore, the feedback and questions during and after each task execution as well as the instructions and observations of the experimenter are gathered. The user actions including their timestamps and the time- and error-rate for the solved task are recorded too. However, with respect to the bias of giving feedback during the task, the possible misinterpretation of the task description and the variance in user skills coupled with a low number of participants, the time- and error-rate have to be interpreted with caution. After solving the full task set, the participant eventually has to rank all UI variants from best to worst. The ranking of all participants of one iteration shows which UI variants support the set of tasks better than others. Furthermore, it may give hint to factors explaining the improvements.

Beside changing UI variants, tasks and their descriptions can be changed or improved between iterations as well, because designing tasks is not straightforward. Too simple tasks, e.g., *identify the largest element*, are not suitable as a real world task for visualization analysis. On the other hand, a complex task is more difficult to explain, may be misinterpreted by the participant, or needs too much time to be solved [Nor06]. Thus, creat-

ing and describing a perfect set of complex tasks from scratch is nearly impossible. To overcome this problem a pilot study is an established way to find weaknesses in tasks and their descriptions. However, the possible task modifications found this way are only a subset and every modification can lead to new weaknesses. Hence, an iterative improvement is a better solution to optimize the tasks. By analyzing the instructions and observations of the experimenter as well as the questions and feedback of the participants the researcher draws several conclusions about the comprehensibility and feasibility. As a result, the complexity of the tasks can be reduced, the descriptions can be remastered, or entire tasks can be replaced.

4.1 Produce Interface Variants

To produce new variants the within-subject design is chosen to encourage the participants to think about the differences between the variants. Therefore, the participants feedback and questions are collected during the whole process and associated to the following categories:

- advantages of variants
- disadvantages of variants
- improvements for variants
- ideas for new variants

By summarizing and interpreting the categorized statements and their rate the researcher draws several conclusions about possible changes. This interpretation process is not of straightforward structure because the researcher and his or her freedom to design the UI is also part of it. For example, the number of gathered disadvantages for one variant may lead the researcher to an idea how to improve this variant to overcome this disadvantages. So the freedom in designing the UI using the qualitative data is the crucial part to find a local optimum in the huge space of possible variants. Nevertheless, the researcher should pay attention to explicitly record his or her decision with respect to further planning of the optimization process. The result of this analysis is an overview following possible changes for the next iteration, weighted by the potential contribution to the effectiveness of the UI:

- adding a complete new variant
- adding an altered existing variant
- adding a combination of existing variants

Attention should be paid to the differences between the variants in one iteration. If they differ in every possible detail of the UI or the ITECs the participants may become confused and the comparison of the variants may not lead to relevant feedback. This would also lead to very long instruction-phases with broad tutorials to explain each variant in detail. Hence, the changes should at least be focused on one category of ITECs, e.g., explore or connect. However, the level of detail in the differences should be taken into account too. The details of the ITECs and their integration into the UI should be investigated after evaluating if and under which conditions a certain ITEC is superior.

Depending on the amount of existing variants and the size of the task set one or more variants can be added for the next iteration. To consider a bigger amount of variants new tasks could be added too, but with respect to the overall length for solving all tasks of the set. On the other side, old variants can be removed if they are ranked low by the participants or have many disadvantages.

4.2 Evaluate Interface Variants

The evaluation of the variants is mainly driven by the user satisfaction, recorded as the ranking from best to worst for all variants after solving the complete task set. To get a ranking for the whole iteration the medians for each variant are computed. An aggregated ranking for all investigated variants in all iterations is built by computing the medians of this iteration rankings, so new variants will not be outnumbered by older ones. This way less effective UI variants are identified and can be excluded from the next iteration. If the process of optimization comes to an end a final variant out of the remaining variants has to be derived. Beside the ranking the circumstances why and when a variant is more effective than another one are also part of this final decision. Therefore, at least all the best ranked variants are investigated further as final candidates by analyzing the advantages and disadvantages as well as comparing the quantitative data of time- and error-rate or the user actions. This may lead to the following four cases:

1. Interpreting the advantages and disadvantages can lead to the conclusion that a final candidate is only superior for a specific type of task. In this case either a new variant should be built upon this insight or, if not possible, all these remaining candidates should be integrated in the final UI with respect to aesthetics of the graphical elements of the UI [ZV14]. Thus the user can decide which variant to use for a task.

2. Computing the relevant statistical parameters of time- and error-rates identifies one final candidate as noticeably superior over the others.

3. One of the candidates has a noticeably lower rate in user actions to solve the tasks than the others. In a

long term usage this candidate should have a higher acceptance by the users.

4. The differences between the candidates are only on a low level of detail, so they could be integrated in the final UI by a configuration option.

If the result of analyzing the final candidates can not be classified as one of these cases either a further investigation by conducting a quantitative study could be done or the researcher eventually has to choose the final UI.

5 DISCUSSION

In this paper we propose some relevant changes to existing UCD processes to reduce the effort for optimizing visualizations. Although we were able to apply the process successfully an evaluation against other evaluation approaches is outstanding due to the required effort. Especially the implementation of the variants is still time-consuming. Since we consider interaction as a crucial factor of success of a visualization we decided against paper prototyping and similar methods. To further increase the efficiency, and thereby being able to evaluate more variants, it is essential to at least partially automate the evaluation of UI variants. However, the current understanding of UI aesthetics is not yet deep enough [ZV14].

Among others, we use quantitative studies to optimize the representation. Even though their number is reduced over time, the first iterations might be even more extensive than existing approaches. However, experience shows that usually many iterations are required and in that case our approach becomes less extensive.

The described approach finds only a local optimum, since it is unfeasible to evaluate all possible variants. This limitation is common to all optimization processes in the area of information visualization. However, our approach comes with a highly efficient evaluation. User studies are used simultaneously for creating and evaluating UI variants in smaller iterations, by analyzing the qualitative data and user ranking. Then the evaluation of the representation is fully automated. Thus, we can investigate a much bigger space to find the local optimum. In turn, the evaluation results are less reliable compared to quantitative studies. Therefore, we propose to finish the optimization process with a controlled experiment to make sure it was successful.

6 RELATED WORK

Several papers address the methodology of evaluating information visualizations [Car08, HWT06, LBIP14, MDF12, MTW+12, SBCS14, TM05]. But they only focus on single evaluations, not on an iterative process as described in this paper. However, iterative optimization is an essential part of UCD. Some authors described such user-centered approaches for information visualization [FZH13, LD11, WKD09]. As we, they try to reduce the resulting effort, e.g., by combining controlled experiments and qualitative methods. Unfortunately, this is achieved at the expense of a drastically reduced interaction evaluation. In contrast, we stress the importance of the interaction but still achieve a reduced effort.

7 CONCLUSION

In this paper, we proposed an improved process for optimizing information visualization regarding readability, comprehensibility, and user satisfaction. Among a heuristic process of finding a local optimum in the huge space of UI variants, we introduced a fully automated evaluation of the representation variants. Although we were able to apply the process successfully an evaluation against other evaluation approaches is outstanding.

8 REFERENCES

[And06] Keith Andrews. Evaluating Information Visualisations. *Proceedings of the 2006 AVI workshop on BEyond time and errors novel evaluation methods for information visualization - BELIV '06*, page 1, 2006.

[Bau15] David Baum. Introducing Aesthetics to Software Visualization. In *Short papers proceedings*, volume 23, page 9, 2015.

[BRSG07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The Aesthetics of Graph Visualization. In *Proceedings of the 2007 Computational Aesthetics in Graphics, Visualization, and Imaging*, 2007.

[Car08] Sheelagh Carpendale. Evaluating Information Visualizations. In *Information Visualization: Human-Centered Issues and Perspectives*, pages 19–45. 2008.

[CC00] Chaomei Chen and Mary P. Czerwinski. Empirical evaluation of information visualizations: an introduction. *International Journal of Human-Computer Studies*, 53(5):631–635, 2000.

[CZ11] P. Caserta and O. Zendra. Visualization of the static aspects of software: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 17(7):913–933, July 2011.

[FZH13] Diana Fernández, Dirk Zeckzer, and José Hernández. A User-Centered Approach for the Design of Interactive Visualizations to Support Urban and Regional Planning. *IADIS International Journal on Computer Science and Information Systems*, 8(2):27–39, 2013.

[Hua14] Weidong Huang. Evaluating Overall Quality of Graph Visualizations Indirectly and Directly. In Weidong Huang, editor, *Handbook of Human Centric Visualization*. 2014.

[HWT06] Nathan Holmberg, Burkhard Wünsche, and Ewan Tempero. A framework for interactive web-based visualization. In *AUIC '06 Proceedings of the 7th Australasian User interface conference*, pages 137–144. Australian Computer Society, Inc., January 2006.

[KSFN08] Andreas Kerren, John T Stasko, Jean-Daniel Fekete, and Chris North. *Information Visualization: Human-Centered Issues and Perspectives*. 2008.

[LBAAL09] H. Ltifi, M. Ben Ayed, A.M. Alimi, and S. Lepreux. Survey of information visualization techniques for exploitation in kdd. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pages 218–225, May 2009.

[LBIP14] Heidi Lam, Enrico Bertini, Petra Isenberg, and Catherine Plaisant. Empirical Studies in Information Visualization: Seven Scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, 2014.

[LCWL14] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[LD11] David Lloyd and Jason Dykes. Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2498–2507, 2011.

[LM08] Heidi Lam and Tamara Munzner. Increasing the utility of quantitative empirical studies for meta-analysis. In *Proceedings of the 2008 Workshop on BEyond Time and Errors: Novel evaLuation Methods for Information Visualization*, BELIV '08, pages 2:1–2:7, New York, NY, USA, 2008. ACM.

[MDF12] Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2536–2545, 2012.

[MTW+12] AV Moere, M Tomitsch, C Wimmer, Boesch C, and T Grechenig. Evaluating the effect of style in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2739–2748, 2012.

[Mun09] Tamara Munzner. A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[Nor06] C. North. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE*, 26(3):6–9, May 2006.

[PAC02] Helen C. Purchase, Jo-Anne Allder, and David Carrington. Graph Layout Aesthetics in UML Diagrams: User Preferences. *Journal of Graph Algorithms and Applications*, 6(3):255–279, 2002.

[Pla04] Catherine Plaisant. The Challenge of Information Visualization Evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, 2004.

[Pur97] Helen C. Purchase. Which Aesthetic Has the Greatest Effect on Human Understanding? In *Proceedings of the 5th International Symposium on Graph Drawing*, 1997.

[PvES06] E Poppe, C van Elzakker, and JE Stoter. Towards a method for automated task-driven generalisation of base maps. *UDMS 2006 - 25th Urban Data Management Symposium*, pages 51–64, 2006.

[SBCS14] Abderrahmane Seriai, Omar Benomar, Benjamin Cerat, and Houari Sahraoui. Validation of Software Visualization Tools: A Systematic Mapping Study. In *2014 Second IEEE Working Conference on Software Visualization*, pages 60–69. IEEE, September 2014.

[TC08] Alfredo R Teyseyre and Marcelo R Campo. An overview of 3D software visualization. *IEEE transactions on visualization and computer graphics*, 15(1):87–105, 2008.

[TM05] Melanie Tory and Torsten Möller. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, 2005.

[vLKS+11] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, and D.W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[WKD09] I Wassink, O Kulyk, and B Van Dijk. Applying a user-centered approach to interactive visualisation design. *Trends in Interactive Visualization*, pages 175–199, 2009.

[WLR11] Richard Wettel, Michele Lanza, and Romain Robbes. Software systems as cities: a controlled experiment. *2011 33rd International Conference on Software Engineering (ICSE)*, pages 551–560, 2011.

[WPCM02] Colin Ware, Helen C. Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.

[YaKSJ07] Ji Soo Yi, Youn ah Kang, J.T. Stasko, and J.A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, Nov 2007.

[ZSAvL14] Elena Zudilova-Seinstra, Tony Adriaansen, and Robert van Liere. *Trends in Interactive Visualization: State-of-the-Art Survey*. Springer, 2014.

[ZV14] Mathieu Zen and Jean Vanderdonckt. Towards an evaluation of graphical user interfaces aesthetics based on metrics. *Proceedings - International Conference on Research Challenges in Information Science*, 2014.

Integrating Depth-HOG and Spatio-Temporal Joints Data for Action Recognition

Noopur Arora
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi-110016, India
mcs142128@cse.iitd.ac.in

Parul Shukla
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi-110016, India
parul@cse.iitd.ac.in

Kanad K. Biswas
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi-110016, India
kkb@cse.iitd.ernet.in

ABSTRACT

In this paper, we propose an approach for human activity recognition using gradient orientation of depth maps and spatio-temporal features from body-joints data. Our approach is based on an amalgamation of key local and global feature descriptors such as spatial pose, temporal variation in 'joints' position and spatio-temporal gradient orientation of depth maps. Additionally, we obtain a motion-induced global shape feature describing the motion dynamics during an action. Feature selection is carried out to select a relevant subset of features for action recognition. The resultant features are evaluated using SVM classifier. We validate our proposed method on our own dataset consisting of 11 classes and a total of 287 videos. We also compare the effectiveness of our method on the MSR-Action3D dataset.

Keywords

Action Recognition, Depth-HOG, Kinect, Body-Joints Data

1 INTRODUCTION

Human action recognition has been an active area of research for over a decade. With the proliferation of online videos and personalized cameras, the task of human action recognition for applications such as content-based video retrieval, surveillance, human-computer interaction has attained newer meanings. Further, the introduction of depth sensors such as Microsoft Kinect has added a new dimension. The depth data available from Kinect consists of depth maps and body-joints data. A number of ways have been used in the literature for action recognition from depth data [Sun11], [Jin12], [Wan12], [WLi10], [BNi11], [Yan12b], [Ore13]. Broadly, these could be categorized as methods that are based on data from depth maps and those, which use joints data.

Li et al.[WLi10] use action graph to model the dynamics of action from depth maps sequences. They use a bag of 3D points to characterize a set of salient postures corresponding to nodes in action graph. Ni et al.[BNi11] use depth-layered multi-channel representation based on spatio-temporal interest points. They propose a multi-modal fusion scheme, developed from spatio-temporal interest points and motion history images, to combine color and depth information. In [Yan12b], the average difference between the depth frames is computed and summarized in a single Depth Motion Maps (DMM), from which Histogram of Oriented Gradients features (HOG) are extracted. Oreifej and Liu [Ore13] construct an activity descriptor called Histogram of Oriented 4D surface normal analogous to the histogram of gradients in color sequences. Jetley and Cuzzolin [Jet14] divide the video into temporally overlapping blocks and generate motion history template (MHT) and binary shape template (BST) for each block. Gradient analysis is performed on MHT and BST to describe motion and shape respectively.

Amongst approaches driven by body-joints data, Sung et al.[Sun11] use features extracted from estimated skeleton and use a two-layered Maximum-Entropy Markov Model (MEMM) where the top layer represents activites and the mid-layer represents sub-activities connected to the corresponding activites in top layer. In [Jin12], the authors propose an encoding scheme to convert skeleton data into symbolic representation and use longest common subsequence for activity recognition. Wang et al.[Wan12] use skeleton data and depth maps to construct novel Local Occupancy Pattern (LOP) feature wherein, each 3D joint is associated with a LOP feature which can be treated as depth appearance of a joint. They further propose fourier

temporal pyramid and use these features in a mining approach to obtain a subset of joints or an actionlet. In [Par15], the authors extract features in spherical coordinate system from body-joints data. The features are represented using bag-of-joint-features (BoJF) model for each joint. To incorporate temporal variations of an action, a hierarchical-temporal histogram (HT-hist) model is used. A new relational geometric feature called Trisarea has been proposed in [Vin15]. It is a pose-based feature defined as the area of trianlge formed by three joints. An approach for reducing pose data over time to histograms of relative location, velocity, and their correlations has been presented in [Ewe15]. Subsequently, the partial least squares have been used to learn a compact and discriminative representation for an action sample.

The use of depth maps has the advantage that cues such as shape and geometry are better represented. Body-joints data, on the other hand, provides pose information which has been known to facilitate action recognition as humans tend to recognize actions easily from a sequence of poses. In this paper, we exploit both the data streams by learning a model based on features extracted from depth maps as well as body-joints data. The features extracted can be categorized as local or global depending on whether the feature descriptors are defined over a local region or the entire video volume. In this paper we propose a novel scheme by integrating both the depth maps and joints data. We estimate Gradient Orientation from depth maps (*depthHOG*) and motion-induced shape (*MIS*) features from depth maps. Further, we augment these features with Relative Joint Distance (*RJD*) and Temporal Joint Distance (*TJD*) features obtained from body-joints data.

The rest of the paper is organized as follows: Section 2 presents the proposed approach. In section 3, we present the experiments and results. Finally, in section 4 we discuss the conclusion and future extensions.

2 PROPOSED APPROACH

In this section, we present our proposed approach based on fusion of key local and global attributes such as pose, temporal joint distance, orientation of gradient and motion information.

2.1 Local Attributes

2.1.1 Spatial Features

It has been widely acknowledged that humans tend to recognize actions easily from a sequence of poses. We use this idea to extract spatial pose-based features, Relative Joint Distance (*RJD*), by computing mean of joint positions in each frame. Let it be denoted by μ^f. Subsequently, in each frame f we compute a Relative Joint Distance (*RJD*) R_j^f of a joint j from the mean as follows:

$$R_j^f = ||p_j^f - \mu^f|| \tag{1}$$

where $p_j^f(x,y,z)$ is the 3D position of a joint j in frame f and μ^f is the mean position of all the given joints in a frame f. We normalize the *RJD* with respect to the height(H) of a person as follows:

$$\hat{R}_j^f = R_j^f / H \tag{2}$$

The *RJD* of each joint over all the frames is concatenated to yield the final spatial descriptor from body-joints data. In particular, we have a 20-dimensional *RJD* feature vector corresponding to the 20 body-joints in a frame. Further, since the execution speeds of an action may vary for different actors, we select N number of frames with a step size of n_f/N and compute *RJD* in these frames only, where n_f is the number of frames in a video. The resultant $N*20$ features capture spatial pose information. However, if an action involves movements such as circular motion of an arm or waving of hands, there will not be significant change in pose. Therefore, there is a need to augment spatial pose features with information from other sources as well.

2.1.2 Temporal Features

We propose to augment spatial pose features with Temporal Joint Distance (*TJD*) features extracted from body-joints data. As with the spatial pose features, we first select N frames from a video sequence of n_f frames. We then compute *TJD* for the selected frames as follows:

$$T_j^f = ||p_j^f - p_j^{f+1}|| \tag{3}$$

Since there are N selected frames, the resultant *TJD* consists of $(N-1)*20$ features.

2.1.3 Spatio-Temporal Features

The *RJD* and *TJD* features are extracted from body-joints data. Additionally, we use depth map sequence to exploit cues such as shape, which are better represented in depth maps. We obtain gradient based spatio-temporal features, henceforth referred to as *depthHOG*. Use of histogram of gradients(HOG) for action recognition has been reported earlier in the literature for RGB data [Sco07], [Kla08], [Per12], [YLi12]. In [Kla08], the authors compute gradients in spatio-temporal pyramid and use regular polyhedrons for quantization of 3D orientations. In [Per12], the authors combine histogram of gradients into orientation tensors per frame.

As a pre-processing step, we normalize the input depth map by performing histogram equalization of intensity values within a person mask on each frame. The

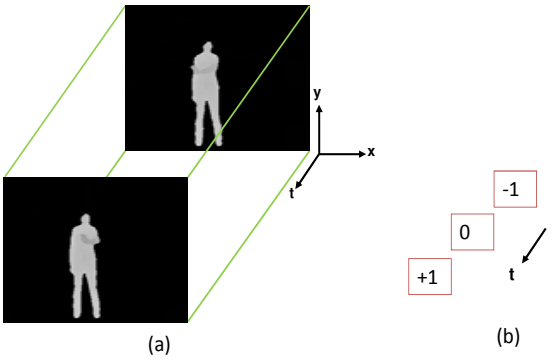


Figure 1: (a)Depth map sequence. (b)Gradient mask for a pixel along temporal domain.

normalization step results in the depth values of person being covered over the entire intensity range. We then compute gradient (G_x, G_y, G_t) of the depth map sequence along the x, y and t directions. Let $D(i, j, f)$ denote the depth value at pixel (i, j) and frame f. The gradients are computed using the following:

$$G_x(i, j, f) = D(i, j+1, f) - D(i, j-1, f) \quad (4)$$

$$G_y(i, j, f) = D(i+1, j, f) - D(i-1, j, f) \quad (5)$$

$$G_t(i, j, f) = D(i, j, f+1) - D(i, j, f-1) \quad (6)$$

Figure 1(a) shows a sample depth map sequence for 'hand wave' action. Figure 1(b) shows the gradient mask across temporal domain. We use the computed gradients (G_x, G_y, G_t) to find local 3D orientations in depth maps. Let $G_x(i, j, f)$ denote the gradient at pixel (i, j) and frame f computed along x direction. Similarly $G_y(i, j, f)$ and $G_t(i, j, f)$ denote the gradients computed along y and t directions respectively. In order to find the local 3D orientation of depth gradients, we convert G_x, G_y, G_t values into spherical coordinates. This results in a gradient magnitude $M(i, j, f)$ and angles $\theta(i, j, f)$ and $\phi(i, j, f)$.

$$M = \sqrt{G_x^2 + G_y^2 + G_t^2}\ ,\ M \geq 0 \quad (7)$$

$$\phi = \arccos(G_t/M)\ ,\ 0 \leq \phi \leq \pi \quad (8)$$

$$\theta = \arctan(G_y/G_x)\ ,\ 0 \leq \theta < 2\pi \quad (9)$$

Although, $\tan(\theta)$ is defined for $-\pi/2 \leq \theta \leq \pi/2$, we map the values in the range $0 \leq \theta < 2\pi$. It may be noted that there is a slight variation from the formulation in [YLi12], in that, their formulation is for RGB data whereas ours is on depth maps. Secondly, in our case, ϕ signifies the orientation of gradient vector with respect to the temporal axis whereas in [YLi12], ϕ is the angle that the gradient vector makes with its projection on the x-y plane.

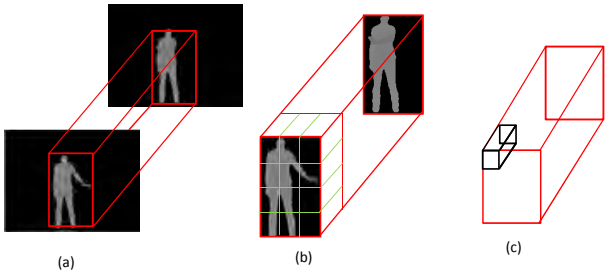


Figure 2: (a)Maximum bounding box for a depth map sequence. (b)Spatio-Temporal grid. (c)Cell in a Spatio-Temporal grid.

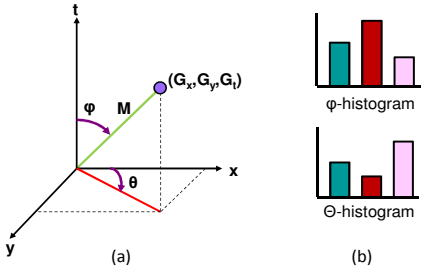


Figure 3: (a)Spherical coordinates for gradient of a pixel. (b)*depthHOG* in a cell.

The aggregation of the orientation values over the depth map sequence is done by dividing the depth map sequence into a spatio-temporal grid. In order to construct such a grid, we consider a Region of Interest (ROI) for a depth map sequence by finding a maximum of all possible bounding boxes (a bounding box contains a person) in a depth map sequence. We then divide this region into a grid consisting of $n_x * n_y$ cells in the spatial domain and n_t cells in the temporal domain. For aggregating the gradient orientations in a cell, we quantize the θ and ϕ angles into n_θ and n_ϕ bins respectively and the bins are weighted according to the gradient magnitude.

Figure 2 illustrates the process of cell creation. Figure 3 illustrates the conversion of pixel gradient into spherical coordiante system and the *depthHOG* as two 1D histograms, namely $\theta - histogram$ and $\phi - histogram$. Each histogram is normalized within a cell. Figure 4(a) and 4(b) illustrates the process of creating angular bins for ϕ and θ. Figure 4(c) and 4(d) illustrate sample histograms in a cell.

The histograms from all the cells are concatenated to give the final *depthHOG* features. The *depthHOG* features are obtained by concatenating $n_x * n_y * n_t$ histograms for both n_θ and n_ϕ bins. A typical choice of the parameters for creating spatio-temporal grid and gradient orientation bins is given as $n_x = 5$, $n_y = 8$, $n_t = 6$, $n_\theta = 12$, $n_\phi = 6$. This would result in 4320 *depthHOG* features.

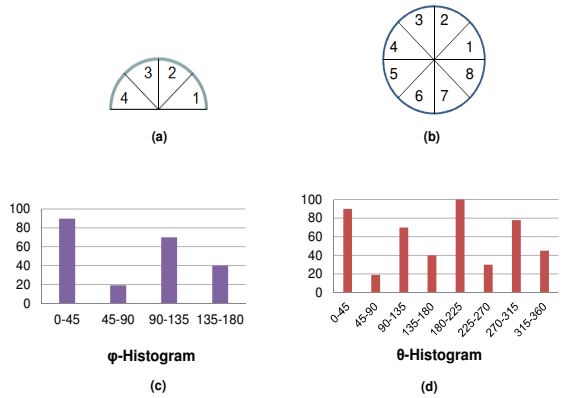


Figure 4: (a)Illustrative example showing 4 angular bins for ϕ. (b)Illustartive example showing 8 angular bins for θ. (c)-(d)Sample $\phi - Histogram$ and $\theta - Histogram$ for a cell.

2.2 Global Attributes

Recent research [Yan12b], [Jet14], suggests that additional body shape and motion information from projections of depth map onto three orthogonal planes can be used to enhance performance of action recognition systems. We use this idea to define a Motion-Induced-Shape (*MIS*) feature. Yang et al. [Yan12b] obtain three 2D maps corresponding to top, front and side views for each depth frame. And for each projected map, obtain motion energy by computing and thresholding the difference between two consecutive maps. This, however, requires one to empirically set a threshold value. We modify this by extracting binary projections along the three directions. In particular, given a depth frame k, we obtain three masks B_k^f, B_k^s and B_k^t corresponding to the three views as:

- *Front view*: $B_k^f(i,j) = 1$, if $D(i,j,k) = z$ and $z > 0$
- *Side view*: $B_k^s(z,j) = 1$, if $D(i,j,k) = z$ and $z > 0$
- *Top view*: $B_k^t(i,z) = 1$, if $D(i,j,k) = z$ and $z > 0$

In all other cases, resultant pixel value will be 0. It may be noted that this procedure is applied only on human silhouette. Obtaining depth information of only human body has been greatly facilitated with devices such as Kinect.

We now aggregate the difference between consecutive binary masks as:

$$S_f(i,j) = \sum_{k=1}^{n_f-1} |B_k^f(i,j) - B_{k+1}^f(i,j)| \qquad (10)$$

$$S_s(i,j) = \sum_{k=1}^{n_f-1} |B_k^s(i,j) - B_{k+1}^s(i,j)| \qquad (11)$$

$$S_t(i,j) = \sum_{k=1}^{n_f-1} |B_k^t(i,j) - B_{k+1}^t(i,j)| \qquad (12)$$

where, $B_k^f(i,j)$ $B_k^s(i,j)$ and $B_k^t(i,j)$ are binary masks corresponding to front, side and top view of depth frame k for pixel (i,j), respectively. Next, we normalize the obtained motion maps as follows:

$$\hat{S}_f(i,j) = \frac{S_f(i,j) - min_f}{max_f - min_f} \qquad (13)$$

where, min_f and max_f are the minimum and maximum pixel values of S_f respectively. Similarly, we normalize S_t and S_s to obtain $\hat{S}_t$ and $\hat{S}_s$. Figure 5 illustrates the normalized motion maps for the 'High arm wave' action.

2.2.1 *Motion-Induced-Shape features*

We obtain *MIS* features by extracting HOG descriptor from the motion maps $\hat{S}_f$, $\hat{S}_t$, $\hat{S}_s$ corresponding to the three views. A typical choice of cell size is $c_x * c_y$ with number of orientation bins as $n_o = 9$ and a block size of $2*2$. c_x and c_y varies for different datasets.

The number of *MIS* features obtained from a single view (say front view) is given as $N_{MIS}^f = n_b * \delta_b * n_o$ where $n_b = n_b^x * n_b^y$ is the number of blocks, $\delta_b = b_x * b_y$ is the block size. Typical value of $b_x = b_y = 2$ indicates that a block consists of $2*2$ cells. If the image is of size $W * H$, then the number of blocks is given as:

$$n_b = \lfloor (\frac{\frac{W}{c_x} - b_x}{(b_x - b_o^x)} + 1) \rfloor * \lfloor (\frac{\frac{H}{c_y} - b_y}{b_y - b_y^o} + 1) \rfloor \qquad (14)$$

where $b_o^x * b_o^y$ denote the block overlap. Typically, $b_o^x = b_o^y = 1$. Likewise, N_{MIS}^s and N_{MIS}^t can be computed from $\hat{S}_s$ and $\hat{S}_t$ for side and top views respectively. Finally, the concatenated MIS descriptors from each of the three views constitute the final *MIS*.

2.3 Classification

The *RJD*, *TJD*, *depthHOG* and *MIS* features from a video are concatenated to form the final feature vector for the corresponding video. We perform classification on the features using SVM [Cha11] with RBF kernel. The resultant feature vector may contain some redundant or irrelevant features leading to large computational load on the classifier. We propose to obtain the most relevant set of features using a feature selection (FS) approach such as RELIEFF [Kon97], [Rob03]. It gives the relative importance of attributes or predictors by keeping into account k nearest neighbors in a class (called as nearest hits) and k nearest neighbors from each of the other classes (called as nearest misses). Prior probability of a class is taken into account while estimating the quality of an attribute.

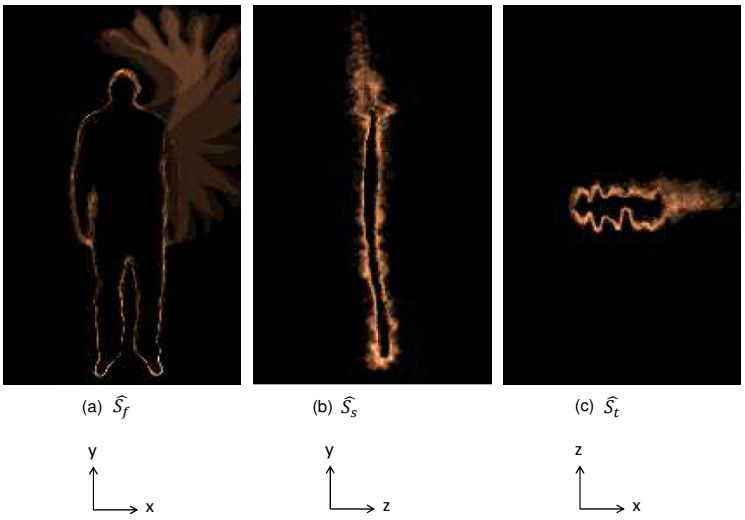


Figure 5: Normalized Motion Maps for High Arm Wave action. (a)Front View (b)Side View (c)Top View

Using RELIEFF we obtain a ranking order of all the features. From the entire set of *ranked* α features, we select a subset of $\hat{\alpha}$ *top* ranked features. We perform classification on the top ranked $\hat{\alpha}$ features using SVM with RBF Kernel. In section 3, we discuss the performance of proposed approach in relation to the number of top ranked features.

3 EXPERIMENTS

In this section, we evaluate the proposed method. We tested our method on the MSR-Action3D dataset [WLi10] and a dataset created by us.

3.1 MSR-Action3D

The MSR-Action3D dataset [WLi10] consists of 20 actions namely 'high arm wave', 'horizontal arm wave', 'hammer', 'hand catch', 'forward punch', 'high throw', 'draw x', 'draw tick', 'draw circle', 'hand clap', 'two hand wave', 'side-boxing', 'bend', 'forward kick', 'side kick', 'jogging', 'tennis swing', 'tennis serve', 'golf swing', 'pick up and throw'. Each action is performed by 10 actors and has a total of 567 depth map sequences as well as body-joints data.

Li et al. [WLi10] divide the 20 actions into three subsets, each having 8 actions as listed in Table 1. The AS1 and AS2 group similar actions with similar movements, while AS3 consists of complex actions. We used the same divisions as well for testing our method. The performance of entire feature set has been compared with that of reduced feature set obtained using RELIEFF in Tables 2 and 3 under 2 scenarios: 'cross-subject'[WLi10] and 'five-fold cross validation'. "Without FS" column refers to the accuracy obtained when the entire set of α features is used. "With FS" column refers to the accuracy obtained using top $\hat{\alpha}$ features. From a total of $\alpha = 11512$, we selected $\hat{\alpha} = 2000$ top ranked features.

In 'cross-subject'[WLi10] setting, half of the subjects are used for training and the remaining are used for testing. In Table 2, we report the accuracy obtained using cross-subject test scenario. We observed an increase in the overall accuracy from 91.28% to 94.61% using feature selection. In 'five-fold cross-validation' the entire dataset is split into five folds and training is done on four folds and tested on remaining fold. This is repeated so that each fold is tested once. The results of the same are reported in Table 3. The accuracy reported is the average over all the folds. We observed an increase in the overall accuracy from 93.73% to 95.92% using feature selection.

Figure 6 illustrates the confusion matrix for AS1, AS2 and AS3 under the 'cross-subject' scenario. It may be observed from fig 6(b) that misclassification occurs mostly for the first five actions since 'draw x', 'draw circle', 'hand catch' involve similar movement of hands. We compare the performance of proposed method ('With FS') with the state-of-the-arts in Table 4.

We also tested our approach on the MSR-Action3D dataset in another scenario wherein the data is not divided into action sets i.e. all the 20 classes were used for evaluation. We obtained an accuracy of 85.09% without feature selection and an accuracy of 87.64% with feature selection in cross-subject test scenario.

3.2 Our Dataset

We created a dataset of depth maps and joints data using Microsoft Kinect to test our proposed approach.

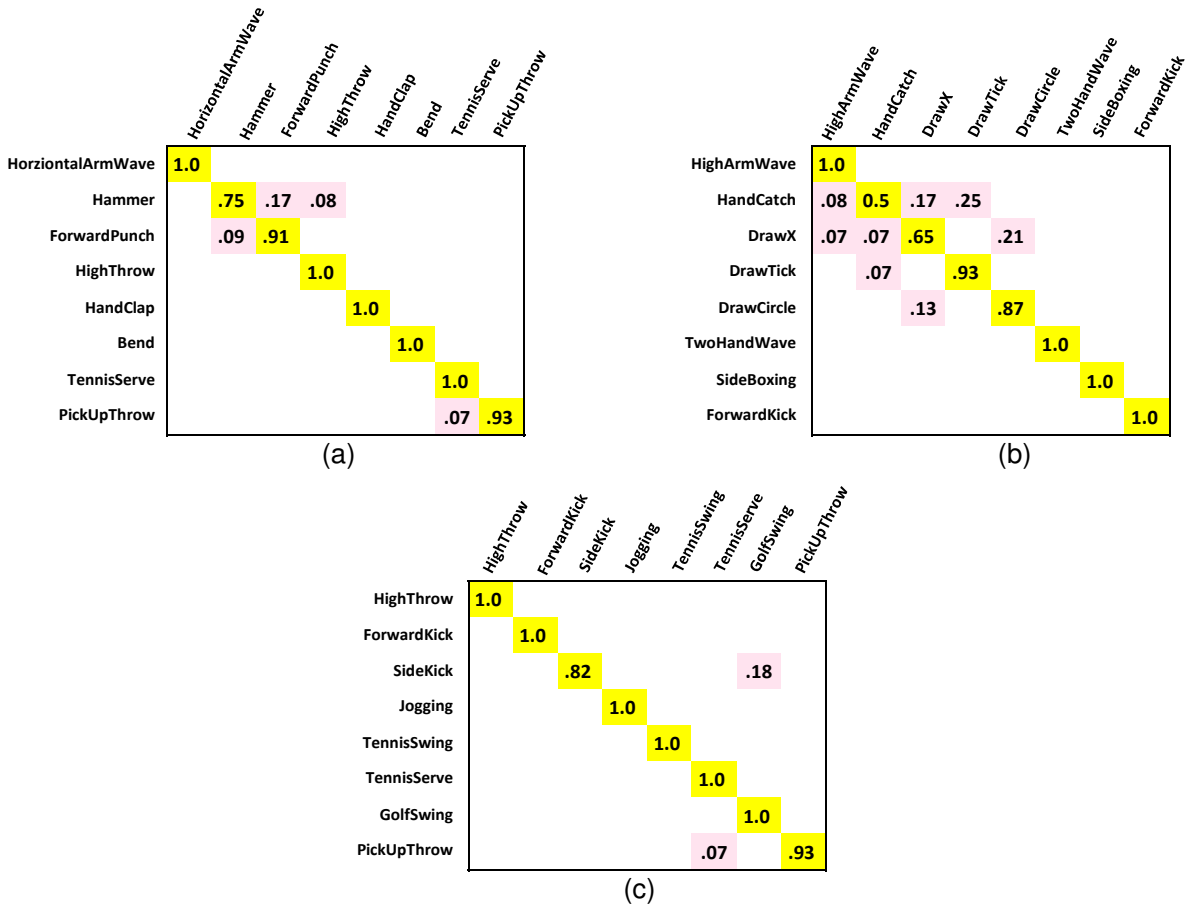


Figure 6: Confusion matrix for MSR Action3D Dataset. (a)AS1 (b)AS2 (c)AS3

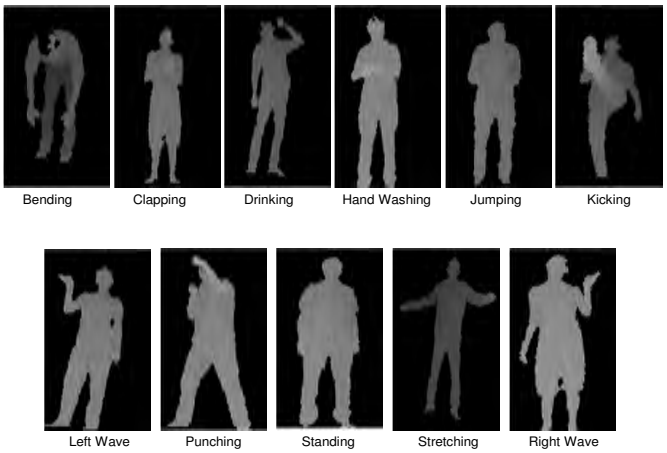


Figure 7: Sample frames from Our Dataset.

Action Set 1 (AS1)	Action Set 2 (AS2)	Action Set 3 (AS3)
Horz. arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup-throw	Side boxing	Pickup-throw

Table 1: The three subsets of actions in MSR Action3D dataset

	Without FS	With FS
AS1	92.45%	96.23%
AS2	84.96%	89.38%
AS3	96.43%	98.21%
Overall	91.28%	**94.61**%

Table 2: Cross-subject accuracy on MSR-Action3D dataset

	Without FS	With FS
AS1	93.36%	96.9%
AS2	90.04%	92.64%
AS3	97.78%	98.23%
Overall	93.73%	95.92%

Table 3: Five-fold cross-validation accuracy on MSR-Action3D dataset

Method	Accuracy
BOP[WLi10]	74.7%
HOJ3D[Xia12]	79.0%
EigenJoints[Yan12a]	82.3%
MHT+BST[Jet14]	83.8%
BoJFH[Par15]	84.5%
GRMD[Sla14]	86.21%
DMM-HOG[Yan12b]	91.63%
Ours	**94.61**%

Table 4: Comparative results on MSR-Action3D dataset in cross-subject scenario

The dataset consists of 11 actions namely 'bending', 'clapping', 'drinking water', 'hand washing', 'jumping', 'kicking', 'left hand wave', 'right hand wave', 'punching', 'standing', 'stretching'. The data set consists of 287 videos where various actions were performed by 13 actors. Figure 7 shows a few sample frames from our dataset.

The total number of features(α) from each video turns out to be 16192 from which we select top 2000 features($\hat{\alpha}$). Table 5 shows the accuracy for 2 testing scenarios: five-fold cross-validation (FFCV) and New Subject(NS). In FFCV scenario, the entire dataset is divided into five folds and training is done on four folds and tested on remaining fold. This is repeated so that each fold is tested once. In 'NS' Test scenario six subjects were chosen for training and the remaining for testing. We observed that the accuracy increased by selecting $\hat{\alpha}$ top ranked feature.

	Without FS	With FS
Five-Fold CV	98.6%	99.3%
New Subject	97.67%	98.45%

Table 5: Results on Our dataset

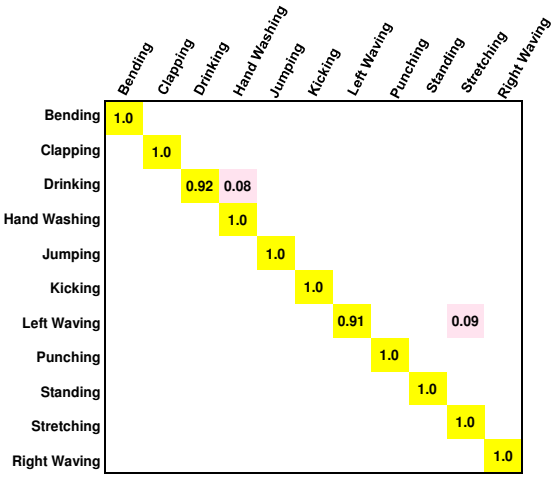


Figure 8: Confusion matrix for our Dataset.

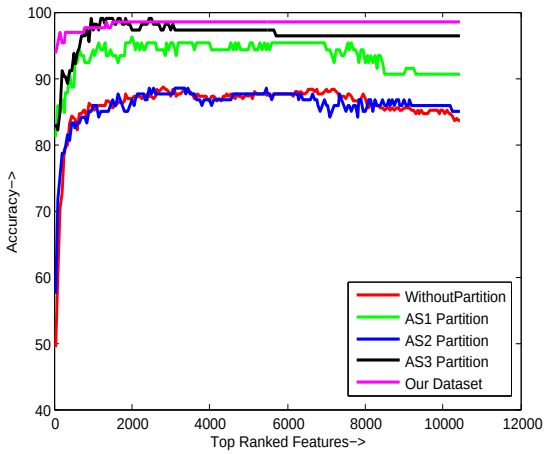


Figure 9: Recognition accuracies using different number of top ranked features.

Figure 8 illustrates the confusion matrix obtained in 'NS' scenario. Figure 9 shows the performance variation with respect to the number of selected top ranked features for MSR Action3D and our dataset. The horizontal axis indicates the number of selected *top* ranked features and the vertical axis indicates the accuracy obtained using the selected features.

4 CONCLUSION

In this paper, we have presented a new approach for action recognition based on fusion of local and global features from depth maps and body-joints data. We have proposed a novel gradient based spatio-temporal feature called as $depthHOG$ and a motion-induced shape

(*MIS*) feature, both extracted from depth maps. Further, we have augmented these features with Relative Joint Distance (*RJD*) and Temporal Joint Distance (*TJD*) feature obtained from body-joints data. We have used RELIEFF to obatin a small but more relevant subset of features from the entire feature pool. Experimental study reveals that the classification accuracy improves when relevant features are used. This further reduces the computational complexity of classification process.

5 REFERENCES

[Cha11] Chang, C. C., and Lin, C. J., LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 2011.

[Ewe15] Eweiwi, A., Cheema, M. S., Bauckhage, C., and Gall, J., Efficient Pose-Based Action Recognition. In *12th Asian Conference on Computer Vision*, Singapore, November 1-5, 2014, Revised Selected Papers, Part V, pages 428–443, 2015.

[Jet14] Jetley, S., and Cuzzolin, F., 3D Activity Recognition Using Motion History and Binary Shape Templates. In *Computer Vision - ACCV 2014 Workshops*, volume 9008 of *Lecture Notes in Computer Science*, pages 129–144, 2014.

[Jin12] Jin, S. Y., and Choi, H. J., Essential body-joint and atomic action detection for human activity recognition using longest common subsequence algorithm. In *Computer Vision - ACCV 2012 Workshops*, volume 7729 of *Lecture Notes in Computer Science*, pages 148–159, 2012.

[Kla08] Kläser, A., Marszałek, M., and Schmid, C., A Spatio-Temporal Descriptor Based on 3D-Gradients. In *British Machine Vision Conference*, pages 995–1004, 2008.

[Kon97] Kononenko, I., Simec, E., and Sikonja, M. R., Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, Volume 7, pages 39–55, 1997.

[YLi12] Li, Y., Sun, T., and Jiang, X., Human Action Recognition Based on Oriented Gradient Histogram of Slide Blocks on Spatio-Temporal Silhouette. In *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 5, No. 3, September, 2012.

[WLi10] Li, W., Zhang, Z., and Liu, Z., Action recognition based on a bag of 3D points. In *CVPR*, 2010.

[BNi11] Ni, B., Wang, G., and Moulin, P., RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *ICCV Workshops*, pages 1147–1153, 2011.

[Ore13] Oreifej, O., and Liu, Z., HON4D: histogram of oriented 4d normals for activity recognition from depth sequences. In *CVPR'13*, pages 716–723, 2013.

[Per12] Perez, E. A., Mota, V. F., Maciel, L. M., Sad, D., and Vieira, M. B., Combining gradient histograms using orientation tensors for human action recognition. In *21st IEEE International Conference on Pattern Recognition (ICPR)*, pages 3460–3463, 2012.

[Rob03] Robnik-Sikonja, M., and Kononenko, I., Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning* , 53, 23–69, 2003

[Sco07] Scovanner, P., Ali, S., and Shah, M., A 3-dimensional sift descriptor and its application to action recognition. In *15th International Conference on Multimedia*, MULTIMEDIA '07, pages 357–360, 2007.

[Par15] Shukla, P., Biswas, K. K., and Kalra, P. K., Bag-of-Features based Activity Classification using Body-joints Data. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, Volume 1, pages 314–322, 2015.

[Sla14] Slama, R., Wannous, H., and Daoudi, M., Grassmannian representation of motion depth for 3d human gesture and action recognition. In *22nd International Conference on Pattern Recognition, ICPR*, pages 3499–3504, 2014.

[Sun11] Sung, J., Ponce, C., Selman, B., and Saxena, A., Human activity detection from RGBD images. In *AAAI workshop on Pattern, Activity and Intent Recognition*, 2011.

[Vin15] Vinagre, M., Aranda, J., and Casals, A., A New Relational Geometric Feature for Human Action Recognition. In *Informatics in Control, Automation and Robotics: 10th International Conference, ICINCO 2013, Iceland, 2013 Revised Selected Papers*, pages 263–278, 2015.

[Wan12] Wang, J., Liu, Z., Wu, Y., and Yuan, J., Mining actionlet ensemble for action recognition with depth cameras. In *CVPR'12*, pages 1290–1297, 2012.

[Xia12] Xia, L., Chen, C., and Aggarwal, J., View Invariant Human Action Recognition Using Histograms of 3D Joints. In *CVPR Workshop*, 2012.

[Yan12a] Yang, X., and Tian, Y., EigenJoints based Action Recognition Using Naive Bayes Nearest Neighbor. In *CVPR Workshop*, 2012.

[Yan12b] Yang, X., Zhang, C., and Tian, Y., Recognizing actions using depth motion maps-based histograms of oriented gradients. In *20th ACM International Conference on Multimedia*, pages 1057–1060, 2012.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Background Modeling using Perception-based Local Pattern

K. L. Chan

Department of Electronic Engineering
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong

itklchan@cityu.edu.hk

ABSTRACT

Background modeling is an important issue in video surveillance. A sophisticated and adaptive background model can be used to detect moving objects which are segregated from the scene in each image frame of the video via the background subtraction process. Many background subtraction methods are proposed for video acquired by a stationary camera, assuming that the background exhibits stationary properties. However, it becomes harder under various dynamic circumstances – illumination changes, background motions, shadows, camera jitter, etc. We propose a versatile background modeling method for representing complex background scenes. The background model is learned from a short sequence of spatio-temporal video data. Each pixel of the background scene is represented by samples of color and local pattern. The local pattern is characterized by perception-inspired features. In order to cater for changes in the scene, the background model is updated along the video based on the background subtraction result. In each new video frame, moving objects are considered as foregrounds which are detected by background subtraction. A pixel is labeled as background when it matches with some samples in the background model. Otherwise, the pixel is labeled as foreground. We propose a novel perception-based matching scheme to estimate the similarity between the pixel and the background model. We test our method using common datasets and achieve better performance than various background subtraction algorithms in some image sequences.

Keywords

Background modeling, Moving object detection, Dynamic background, Background subtraction, Local pattern

1. INTRODUCTION

One of the most challenging problems in computer vision is to detect and recognize moving objects such as humans or vehicles in complex environments automatically. Video surveillance [Hsi08a] is obviously one well-known application. For instance, automatic video surveillance systems for human motion monitoring typically consist of the human detection, tracking of targets along the video sequence, and inference of the motion. Besides, other areas such as gait analysis [Cun03a] and video segmentation and retrieval [Lu04a], also benefit from the advance in moving object detection research. The detection of moving objects as foregrounds in the video is the first key problem. To detect moving targets, one common approach is to create a model representing the background scene. The background model is used to detect moving objects by the background subtraction process. At the same time, the background model is updated to cater for the changes in the scene. In each image frame of the video, the background subtraction process is to find out those pixels that are similar to the background. The pixels that are not similar to the background belong to the moving objects (foregrounds). The process involves matching of the pixels with the background model.

Background model can be created and updated from the video. One common assumption is that the video is acquired by a fixed camera and the scene is stationary or changes slowly. However, the scene is not always static. The captured environment can have dynamic elements such as illumination changes, waving trees, water, etc. Strong wind can cause camera jitter. Therefore, sophisticated background modeling methods are proposed for tackling scene variations and background movements. Sobral and Vacavant [Sob14a] presented a recent review and evaluation of 29 background subtraction methods. One approach is to represent the background scene by parametric model. For instance, pixelwise

background color can be modeled by Gaussian distribution. Stauffer *et al.* [Sta00a] proposed the modeling of background colors using mixture of Gaussian (MOG) distributions as individual scene pixels may exhibit multiple colors because of background motions or illumination changes. Background model is initialized using an EM algorithm. Pixel values that do not match any of the background distributions are regarded as foreground. Parameters of the MOG model are updated after foreground detection. Since its introduction, MOG has gained widespread popularity and inspired many improvements. For instance, in contrast with a fixed number of Gaussians in the original MOG model, Zivkovic [Ziv04a] proposed an algorithm for selecting the number of Gaussian distributions using the Dirichlet prior. A comprehensive survey on the improvements of MOG model can be found in [Bou08a].

Another approach is to create non-parametric background model. This category of background subtraction methods does not assume the pdf of the scene follow a known parametric form. Elgammal *et al.* [Elg02a] proposed an algorithm for estimating the pdf directly from previous pixels using kernel estimator. Barnich and Van Droogenbroeck [Bar09a, Bar11a] proposed a sample-based background subtraction algorithm called ViBe. Background model is initialized by randomly sampling of pixels on the first image frame. Pixel of the new image frame is classified as background when some samples intersecting the sphere of the pixel. A random policy is also employed for updating the background model at the pixel location and its neighbor. Hofmann *et al.* [Hof12a] proposed a similar non-parametric sample-based background subtraction method. The method can adaptively adjust the foreground decision threshold and model update rate along the video sequence. Haines and Xiang [Hai14a] presented a non-parametric background modeling method based on Dirichlet process Gaussian mixture models. Gao *et al.* [Gao14a] and Liu *et al.* [Liu15a] regarded the observed video frames as a matrix, which can be decomposed into a low-rank matrix of background and a structured sparse matrix of foreground.

Recently, methods for modeling background scene by local pattern are proposed. Heikkilä and Pietikäinen [Hei06a] proposed to model the background of a pixel by local binary pattern (LBP) histograms estimated around that pixel. Liao *et al.* [Lia10a] proposed the scale invariant local ternary pattern (SILTP) which can tackle illumination variations. St-Charles *et al.* [Stc15a] proposed a pixelwise background modeling using local binary similarity pattern (LBSP) estimated in the spatio-temporal domain. Their method outperforms 32 state-of-the-art methods on the ChangeDetection.net dataset [Goy12a, Wan14a].

In this work, we have two contributions. First, we propose a novel perception-based local pattern which can be used effectively to characterize various dynamic circumstances in the scene. Second, we propose a novel scheme to estimate the similarity between new pixel and the background model for classifying the pixel. The background model and the pixel classification are incorporated into the background subtraction method for moving object detection. The background subtraction result is used to update the background model.

2. BACKGROUND MODEL INITIALIZATION

It is common that the background model is created from the video. The modeling method must be versatile since various scene complications may be encountered. We consider that the feature representing the background scene is an important factor. To make the modeling method generic, there should be as few tunable parameters as possible.

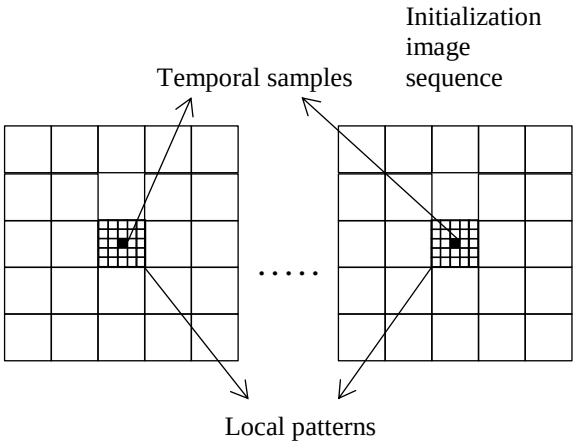


Figure 1. Spatio-temporal sampling of background pixels.

In sample-based background subtraction, the background model is generated by taking previous samples at the same pixel position like [Elg02a], or taking random samples on the first image frame [Bar09a, Bar11a]. We observed various challenges in real scenes. Dynamic background elements such as tree and water produce many false positive errors. Camera jitter also produces false positive errors. It is because the background model does not contain sufficient and representative samples. We propose to take samples from the spatio-temporal domain. As shown in Figure 1, in background model initialization, a number of image frames are used. At a given pixel location (the dark pixels in Figure 1), colors of all the samples (temporal samples) at the same position are entered into the background model

for that pixel. In addition, a block is defined centered at that pixel and local pattern feature is extracted from this block of pixels. All spatio-temporal local patterns, sampled from all pixels of a short initialization image sequence, are also entered into the background model. We have performed experimentations and finally fixed the number of initialization image frames as 30 and the block size as 5 x 5 pixels as shown in Figure 1. Static

background can be represented by the temporal samples while dynamic background can be represented by the spatio-temporal local patterns. In case there are moving objects in the initialization image frames, the model still contains background samples as far as the objects are not stationary. The effectiveness of the background model can be seen in the results from camera jitter videos in section 4.

In dynamic scenes, the colors of background elements can vary due to illumination change. The variations of colors must be allowed in matching the new pixel with the background model. Inspired by the perception-inspired confidence interval [Haq13a], we propose a novel local pattern that can cater for color variations. The confidence interval of a sample having a color component value c is defined as ($c - d$, $c + d$). According to Weber's law [Gon10a], d depends on the perceptual characteristics of c. That is, d should be small for darker color and large for brighter color. The perception-based linear relationship is formulated as

$$d = 0.11 * c \quad (1)$$

Each pixel of the block (except the center pixel) is compared with the center pixel. If its color is outside the confidence interval of the center pixel, its feature value f is set equal to

$$f = b_{half} - d_{city} + 1 \quad (2)$$

where d_{city} is the city-block distance between a given pixel of the block and the center pixel, b_{half} is the half size of the block. If its color is within the confidence interval of the center pixel, its feature value is 0. Therefore, neighbor closer to the center pixel will contribute a larger feature value if they are perceptually different. Different neighbor farther from the center pixel will contribute a smaller feature value. Finally all feature values of the block are summed to form the pattern value for the center pixel. Figure 2a illustrates the formation of a local pattern for a block of 3 x 3 pixels. The first row shows the formation of LBP for a noise-free image. The second row indicates that LBP is not robust to random noise in the image. The third row also shows that LBP cannot keep its invariance against scale transform. Figure 2b illustrates the formation of perception-based local pattern under the same circumstances. The confidence interval for the patterns in the first and second row is (56.96, 71.04). The confidence interval for the pattern in the third row is (113.92, 142.08). It can be seen that perception-based local pattern is robust against random noise and scale transform. Its pattern value is equal to 4.

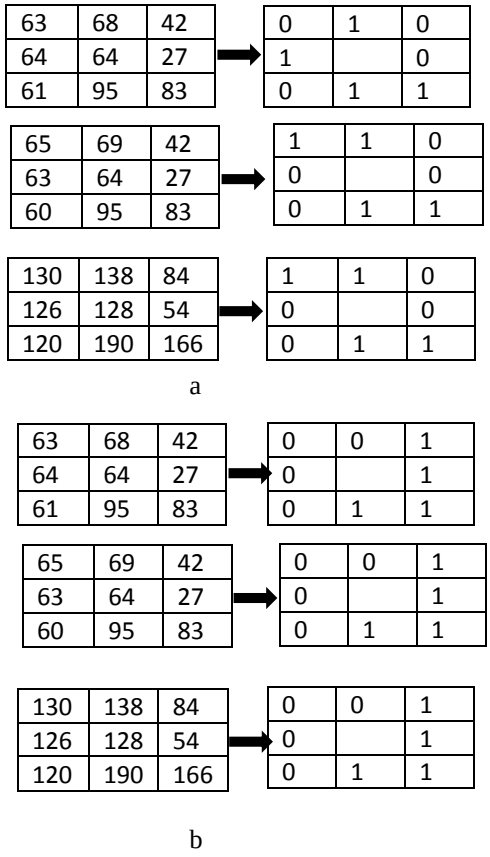


Figure 2. Formation of local pattern: (a) LBP, (b) perception-based local pattern.

We observed that the choice of color model can have significant impact on the accuracy of moving object detection. We used invariant color feature to represent the color of the pixel. In our method, we adopted the $c_1c_2c_3$ normalized color model [Gev99a].

$$c_1 = \arctan\frac{R}{\max(G, B)} \quad (3)$$

$$c_2 = \arctan\frac{G}{\max(R, B)} \quad (4)$$

$$c_3 = \arctan\frac{B}{\max(R, G)} \quad (5)$$

3. MOVING OBJECT DETECTION AND BACKGROUND MODEL UPDATING

Figure 3 illustrates the framework of our moving object detection method. The background model is initialized using 30 initial image frames of the video

as mentioned in the previous section. In the background/foreground segmentation, all pixels of the current image frame are classified as background or foreground. Since we have generated a strong background model that characterizes the spatial and temporal variations of background colors, we adopt a conservative policy in the joint background/foreground segmentation. If all color component values of the pixel match with some temporal color samples or spatio-temporal local patterns of the background model, the pixel is labeled as background. Otherwise, it is labeled as foreground.

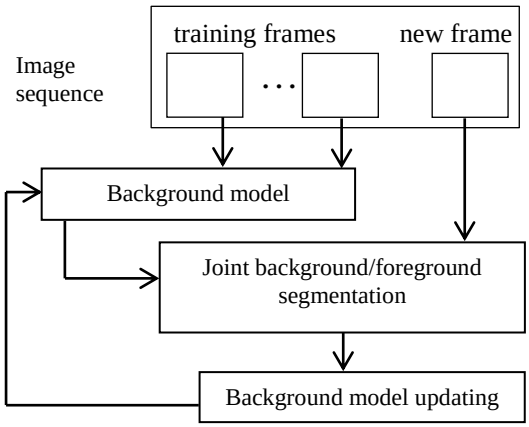


Figure 3. Overview of our moving objects detection method.

We propose a novel scheme to estimate the similarity between the pixel and the background model which strikes for balance between efficiency and perceptual accuracy. First, the pixel is compared with the temporal color samples of the background model. The perception-based confidence interval of the pixel is defined. Once two temporal color samples in the background model are found fall within the confidence interval, the pixel is labeled as background. In static scene, the background subtraction can be accomplished quickly by this process. In dynamic scene, it may not be possible to find similar color samples at the same spatial location along the temporal domain. Then, the pixel is compared with the spatio-temporal local patterns in the background model. A block with this pixel at the center is defined. Pattern values for this pixel are calculated using the same method as mentioned in the previous section. Local pattern of the pixel is compared with the patterns stored in the background model. We define a spatio-temporal search space of 11 x 11 pixels x 30 frames. Two patterns are considered similar if the absolute difference of their pattern values is ≤ a tolerance value. We fixed the tolerance value to 3. If two patterns in the background model match with the local pattern of the pixel, the pixel is labeled as background. Otherwise, the pixel is labeled as foreground. The algorithm of background subtraction is shown below.

Algorithm – background subtraction

For each new pixel
  Define perception-based confidence interval
  Search temporal color samples
  If number of matches = 2
    Label pixel as background
    Step over to the next pixel
  Else
    Calculate perception-based local pattern
    Search spatio-temporal local patterns
    If number of matches = 2
      Label pixel as background
      Step over to the next pixel
    Else
      Label pixel as foreground

In the background model updating, the total number of color samples and local patterns will remain the same. If the new pixel matches with the temporal color samples, one temporal color sample will be updated by the following equation

$$c_b^{new} = (1-\alpha)c_b^{old} + \alpha c_p \quad (6)$$

where c_p is the color of the new pixel, c_b is the matched temporal color. We set α equal to 0.05. If the local pattern of the new pixel matches with the patterns of the background model, one local pattern will be updated by the following equation

$$l_b^{new} = (1-\alpha)l_b^{old} + \alpha l_p \quad (7)$$

where l_p is the local pattern value of the new pixel, l_b is the matched local pattern value in the background model.

The use of chromaticity in matching the pixel with background model means the background/foreground segmentation is robust to gradual illumination change. We also observe that cast shadow is more likely to be classified as background rather than foreground by using chromaticity. We use the same set up in the experimentation. There are no tunable parameters.

4. RESULT

We implement our method using MATLAB and run on a 2.1 GHz PC with 1 Gbyte memory. For a low-resolution image frame of 320 x 240 pixels, the computation time per image frame is about 5 seconds. In the first experimentation, we evaluate our method quantitatively in terms of Recall (Re),

Precision (Pr), F-Measure (F1), False Positive Rate (FPR), and False Negative Rate (FNR) using the Change Detection dataset [Goy12a]. Recall gives the ratio of detected true positive pixels (TP) to total number of foreground pixels present in the ground truth which is the sum of true positive and false negative pixels (FN). Precision gives the ratio of detected true positive pixels to total number of foreground pixels detected by the method which is the sum of true positive and false positive pixels (FP). F-Measure is the weighted harmonic mean of Precision and Recall. It can be used to rank different methods. The higher the value of Re, Pr, and F1, the better is the accuracy. The lower the value of FPR and FNR, the better is the accuracy.

Table 1 shows the average F1 of our method and some well-known parametric and non-parametric background subtraction algorithms obtained from 5 categories of video (baseline - B, dynamic background - DB, camera jitter - CJ, intermittent object motion - IOM, shadow - S), containing 26 image sequences of 47,040 image frames. The best result in a given column is highlighted. No method can achieve the best result in all categories. GMM [Sta00a], KDE [Elg02a] and ViBe [Bar11a] can achieve the best F1 in one category. Our method can achieve the best F1 in two categories of dynamic background and camera jitter, and the results in other categories are close to the best F1.

	B	DB	CJ	IOM	S
GMM	0.825	0.633	0.597	**0.520**	0.716
KDE	**0.909**	0.596	0.572	0.409	0.766
ViBe	0.866	0.459	0.569	0.488	**0.798**
Our method	0.884	**0.635**	**0.671**	0.475	0.712

Table 1. Average F1 of various methods on the Change Detection dataset

We then present a detail comparison of our method with ViBe. We select ViBe because it was showed that ViBe performs better than many state-of-the-art parametric and non-parametric algorithms such as [Ziv04a]. Tables 2 and 3 show the results of our method and ViBe on the dynamic background category respectively. In the tables, the best average results are highlighted. There are six image sequences (boats, canoe, fall, fountain01, fountain02, overpass). The videos contain strong background motions such as moving water and tree shaken by the wind. Our method can achieve higher F1 than ViBe in all image sequences. Our method can achieve better result than ViBe in 3 out of 5 average quantitative measures. Tables 4 and 5 show the results of our method and ViBe on the camera jitter category respectively. There are four image sequences (sidewalk, boulevard, traffic, badminton). The videos were captured by vibrating cameras. All videos are very challenging. Our method can achieve higher F1 than ViBe in 3 out of 4 image sequences. Our method can achieve better result than ViBe in 3 out of 5 average quantitative measures.

Sequence	Re	Pr	F1	FPR	FNR
boats	0.682	0.842	0.754	0.001	0.318
canoe	0.856	0.915	0.885	0.003	0.144
fall	0.713	0.546	0.618	0.011	0.287
fountain01	0.339	0.133	0.191	0.002	0.661
fountain02	0.733	0.470	0.573	0.002	0.267
overpass	0.805	0.780	0.792	0.003	0.195
Average	0.688	**0.614**	**0.635**	**0.004**	0.312

Table 2. Results of our method – dynamic background

Sequence	Re	Pr	F1	FPR	FNR
boats	0.528	0.107	0.178	0.020	0.472
canoe	0.897	0.694	0.783	0.014	0.103
fall	0.833	0.342	0.484	0.036	0.168
fountain01	0.580	0.032	0.061	0.008	0.420
fountain02	0.822	0.428	0.563	0.002	0.179
overpass	0.798	0.600	0.685	0.005	0.202
Average	**0.743**	0.367	0.459	0.014	**0.257**

Table 3. Results of ViBe – dynamic background

Sequence	Re	Pr	F1	FPR	FNR
sidewalk	0.405	0.837	0.546	0.002	0.595
boulevard	0.684	0.867	0.765	0.005	0.316
traffic	0.589	0.736	0.654	0.014	0.411
badminton	0.587	0.927	0.719	0.002	0.413
Average	0.566	**0.842**	**0.671**	**0.006**	0.434

Table 4. Results of our method – camera jitter

Sequence	Re	Pr	F1	FPR	FNR
sidewalk	0.518	0.279	0.363	0.027	0.482
boulevard	0.782	0.444	0.566	0.037	0.219
traffic	0.851	0.559	0.675	0.039	0.149
badminton	0.835	0.562	0.672	0.017	0.166
Average	**0.746**	0.461	0.569	0.030	**0.254**

Table 5. Results of ViBe – camera jitter

Figure 4 shows some visual results from the dynamic background category. The first column shows the original image frames and the results obtained by ViBe. The second column shows the results obtained by our method. The third column shows the corresponding ground truths. The ground truth images contain 5 labels (static, hard shadow, outside region of interest, unknown motion, motion). It can be seen that ViBe produces more false positive errors than our method in all image sequences. ViBe may also produce many false negative errors (see results of boats and overpass). From the figure, it can be seen that our method produces balanced Recall and Precision. That is why our method can achieve higher F1 in all image sequences. Figure 5 shows the visual results from the camera jitter category. Again, ViBe produces more false positive errors than our method in all image sequences. In sidewalk, the stationary human and crossing are erroneously detected as foreground by ViBe. Our method only produces minimal scattered false positive errors in the stationary human, while the crossing is correctly identified as background. In the badminton, the players appear at the beginning of the image sequence. Unfortunately, ViBe erroneously detects those players when they already moved to different places along the image sequence. As shown in the figure, our method can detect the correct number of players.

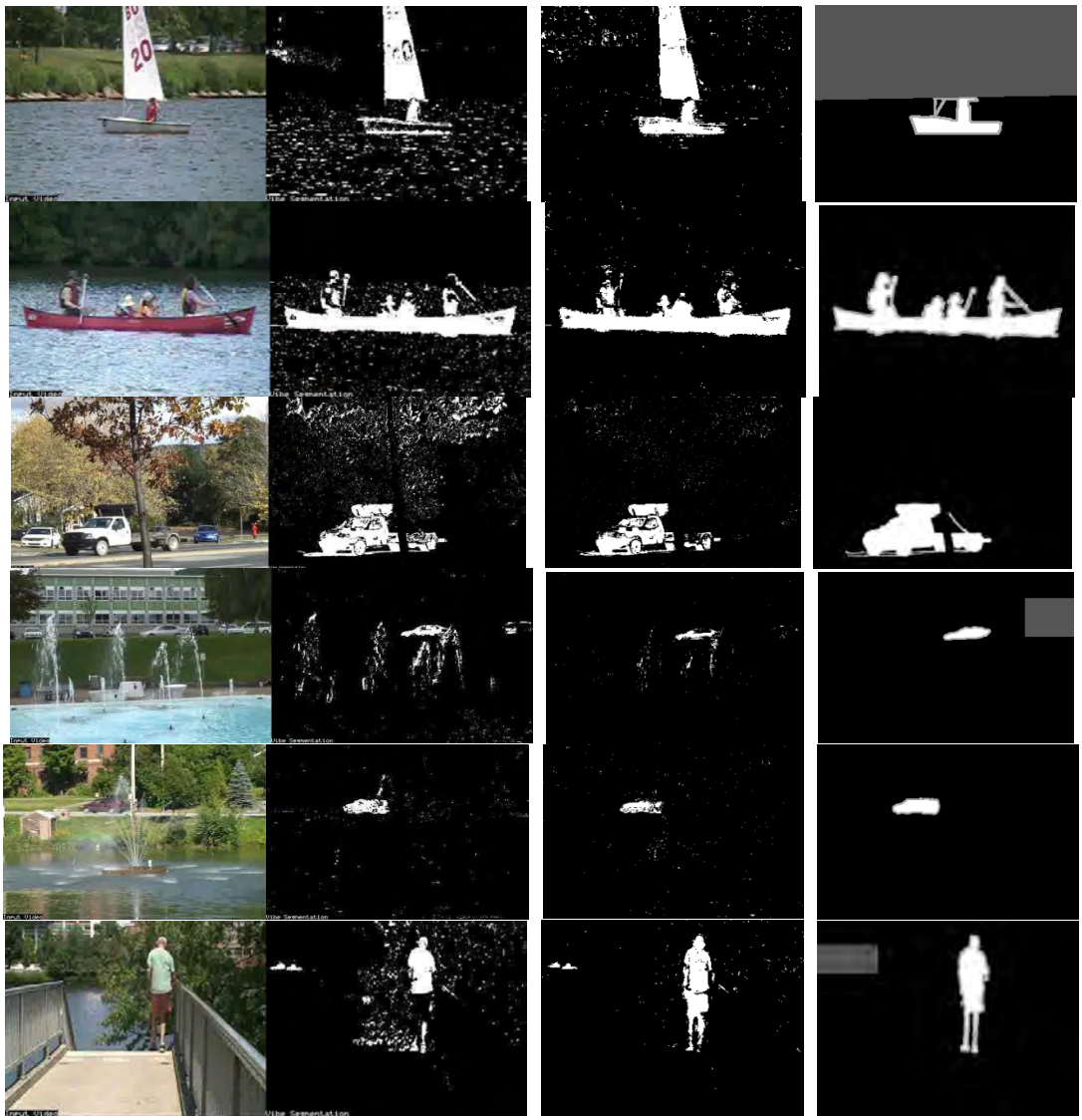


Figure 4. Background subtraction results from the Change Detection dataset dynamic background category – original image frames and results obtained by ViBe (first column), results obtained by our method (second column), ground truths (last column).

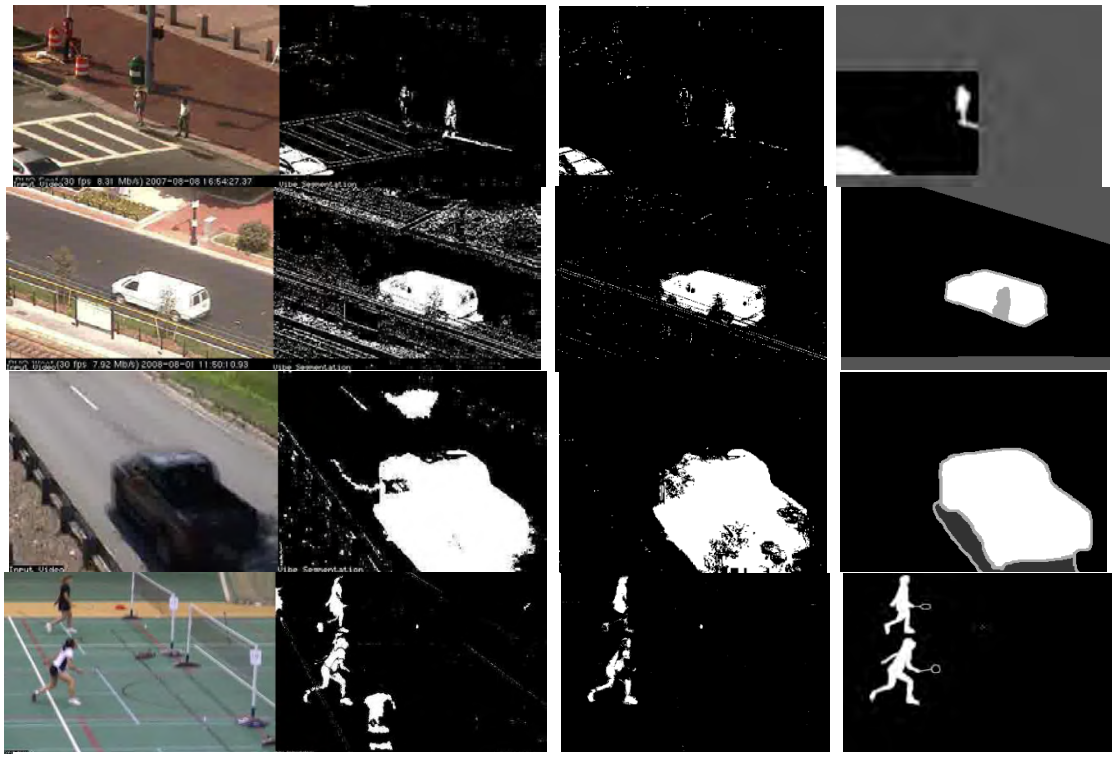


Figure 5. Background subtraction results from the Change Detection dataset camera jitter category – original image frames and results obtained by ViBe (first column), results obtained by our method (second column), ground truths (last column).

In the second experimentation, we compare our method with ViBe and some local pattern based background subtraction algorithms (blockwise LBP – LBP-B [Hei04a], pixelwise LBP – LBP-P [Hei06a]) using the STAR dataset [Li03a]. Table 6 shows the F1 of 9 video sequences. The superiority of local pattern based background model over sampled-based background model can be seen. Our method can achieve the best F1 in 3 video sequences, and the average F1 is second to LBP-P.

Sequence	**LBP-B**	**LBP-P**	**ViBe**	**Our method**
Airport Hall	0.477	**0.503**	0.496	0.429
Bootstrap	0.528	0.520	0.514	**0.569**
Curtain	0.661	0.714	0.775	**0.800**
Escalator	**0.591**	0.539	0.445	0.380
Fountain	0.705	**0.753**	0.425	0.484
Shopping Mall	0.547	**0.629**	0.522	0.548
Lobby	0.503	**0.523**	0.029	0.448
Trees	**0.629**	0.606	0.345	0.600
Water Surface	0.768	0.822	0.801	**0.878**
Average	0.587	**0.635**	0.444	0.600

Table 6. F1 of various methods on the STAR dataset

5. CONCLUSION

We propose a method for the detection of moving objects in video. The background model is represented by samples of color and perception-based local patterns. In moving object detection, each pixel of the current image frame is classified as background if it matches with the background model. Otherwise, the pixel is classified as foreground. This is achieved by our proposed perception-based matching scheme to estimate the similarity between the pixel and the background model. We test and compare our method with various well-known background subtraction algorithms using challenging video datasets. The quantitative measures and visual results show that our method can achieve better performance in some image sequences.

6. REFERENCES

[Bar09a] Barnich, O., and Van Droogenbroeck, M. ViBe: a powerful random technique to estimate the background in video sequences. Proc. of Int. Conf. on Acoustics, Speech and Signal Processing, pp. 945-948, 2009.

[Bar11a] Barnich, O., and Van Droogenbroeck, M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. on Image Processing, Vol. 20, No. 6, pp. 1709-1724, 2011.

[Bou08a] Bouwmans, T., El Baf, F., and Vachon, B. Background modeling using mixture of Gaussians for foreground detection – a survey. Recent Patents on Computer Science, Vol. 1, pp. 219-237, 2008.

[Cun03a] Cunado, D., Nixon, M.S., and Carter, J.N. Automatic extraction and description of human gait models for recognition purposes. Computer Vision and Image Understanding, Vol. 90, pp. 1-41, 2003.

[Elg02a] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. of IEEE, Vol. 90, No. 7, pp. 1151-1163, 2002.

[Gao14a] Gao, Z., Cheong, L.-F., Wang, Y.-X. Block-sparse RPCA for salient motion detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 36, No. 10, pp. 1975-1987, 2014.

[Gev99a] Gevers, T., and Smeulders, A.W.M. Color based object recognition. Pattern Recognition, Vol. 32, pp. 453-464, 1999.

[Gon10a] Gonzalez, R.C., and Woods, R.E. Digital Image Processing. Pearson/Prentice Hall, 2010.

[Goy12a] Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., and Ishwar, P. Changedetection.net: a new change detection benchmark dataset. Proc. of IEEE Workshop on Change Detection at IEEE Conf. on Computer Vision and Pattern Recognition, pp. 16-21, 2012.

[Hai14a] Haines, T.S.F., and Xiang, T. Background subtraction with Dirichlet process mixture models. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 36, No. 4, pp. 670-683, 2014.

[Haq13a] Haque, M., and Murshed, M. Perception-inspired background subtraction. IEEE Trans. on Circuits and Systems for Video Technology, Vol. 23, No. 12, pp. 2127-2140, 2013.

[Hei04a] Heikkilä, M., Pietikäinen, M., and Heikkilä, J. A texture-based method for detecting moving objects. Proc. of British Machine Vision Conf., pp. 187-196, 2004.

[Hei06a] Heikkilä, M., and Pietikäinen, M. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 28, No. 4, pp. 657-662, 2006.

[Hof12a] Hofmann, M., Tiefenbacher, P., and Rigoll, G. Background segmentation with feedback: the Pixel-Based Adaptive Segmenter. Proc. of IEEE Workshop on Change Detection at IEEE Conf. on Computer Vision and Pattern Recognition, pp. 38-43, 2012.

[Hsi08a] Hsieh, J.–W., Hsu, Y.–T., Liao, H.–Y.M., and Chen, C.–C. Video-based human movement analysis and its application to surveillance systems. IEEE Trans. on Multimedia, Vol. 10, No. 3, pp. 372-384, 2008.

[Li03a] Li, L., Huang, W., Gu, I., and Tian, Q. Foreground object detection from videos containing complex background. Proc. of ACM Int. Conf. on Multimedia, pp. 2-10, 2003.

[Lia10a] Liao, S., Zhao, G., Kellokumpu, V., Pietikäinen, M., and Li, S.Z. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1301-1306, 2010.

[Liu15a] Liu, X., Zhao, G., Yao, J., Qi, C. Background subtraction based on low-rank and structured sparse decomposition. IEEE Trans. on Image Processing, Vol. 24, No. 8, pp. 2502-2514, 2015.

[Lu04a] Lu, C.M., and Ferrier, N.J. Repetitive motion analysis: segmentation and event classification. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26, No. 2, pp. 258-263, 2004.

[Sob14a] Sobral, A., and Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Computer Vision and Image Understanding, Vol. 122, pp. 4-21, 2014.

[Sta00a] Stauffer, C., and Grimson, W.E.L. Learning patterns of activity using real-time tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 747-757, 2000.

[Stc15a] St-Charles, P.-L., Bilodeau, G.-A., and Bergevin, R. SuBSENSE: a universal change detection method with local adaptive sensitivity. IEEE Trans. on Image Processing, Vol. 24, No. 1, pp. 359-373, 2015.

[Wan14a] Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., and Ishwar, P. CDnet 2014: an expanded change detection benchmark dataset. Proc. of IEEE Workshop on Change Detection at IEEE Conf. on Computer Vision and Pattern Recognition, pp. 387-394, 2014.

[Ziv04a] Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. Proc. of Int. Conf. on Pattern Recognition, pp. 28-31, 2004.

Toward a Computational Model and Decision Support System for Reducing Errors in Pharmaceutical Packaging

Carson Quigley
Biomedical Engineering
Bucknell University
17837, Lewisburg, PA
carson.quigley@bucknell.edu

Steven Shooter
Mechanical Engineering
Bucknell University
17837, Lewisburg, PA
shooter@bucknell.edu

Aaron Mitchel
Psychology
Bucknell University
17837, Lewisburg, PA
aaron.mitchel@bucknell.edu

Scarlett Miller
School of Design
Penn State University
16801, University Park, PA
scarlettmiller@psu.edu

Dean Parry
Pharmacies
Geisinger Health System
17822, Danville, PA
dparry@geisinger.edu

ABSTRACT

The US Institute of Medicine reports that one medication error occurs *per patient per day* in hospital care, and other studies indicate that medication administration errors attributed to packaging and/or labeling confusion can be as high as 33%. While many engineered products have identifiable features that help establish commonality and differentiation within a product family, vital features of consumable products such as medications are often not readily apparent in their physical form. As a result, caregivers must rely on the labeling and packaging to effectively determine the contents. Adverse Drug Events (ADEs) are the most common category of medical errors and include wrong drug, wrong dose, wrong route of administration, and wrong patient. It is estimated that in the US each year, medication errors harm at least 1.5 million people, resulting in 106,000 deaths. Computational models and associated decision support systems have the potential to improve pharmaceutical delivery safety through informed design of packaging features and enhanced situational awareness and decision-making during drug identification and administration. Past research has led to the formulation of measures for representing the degree of commonality and differentiation of packaging features in pharmaceutical families or versus look-alike drugs. Preliminary studies have validated these measures of feature prominence based on feature size and location. This paper describes a study using eye tracking to evaluate gaze patterns and further validate these measures. The results support the measures and indicate that increased commonality of features results in shorter reaction times, but also shorter fixation times. These results have implications in the formulation of a resulting decision support system.

Keywords

Pattern Recognition, Pharmaceutical Safety, Product Family Planning

1. INTRODUCTION

U.S. Pharmacia estimates that there are approximately 62.9 million medication dispensing mistakes a year in hospitals and pharmacies nationwide (Hicks, 2008), with just over three million of these mistakes considered to be medically significant. These significant mistakes can result from misreading labels, misprescribing, giving the wrong dose, or incomplete documentation. Classified as ADEs or "Adverse Drug Events," these events are defined and classified as "an injury due to medication management rather than the underlying condition of the patient" (Aspden, 2007). When in a hospital, it is estimated that a patient is on the receiving end of these mistakes an average of once per day, which can add up to $6,000 to a medical bill (Aspden, 2007). These errors can and do have detrimental effects to the patients' health and finances. Aside from the individual and familial effects that are seen from decreased health and death from illness or an ADE, there are large societal costs. These mistakes cost the nurses, the pharmacy and the health insurance providers. The accrued total of these mistakes must be distributed to cover the end cost to the patient as well as the healthcare companies. To do this, these costs are implemented throughout various industries, costing society as a whole not just the ones being directly

impacted by the mistakes. Often these mistakes occur because of the lack of consistency and regulation among package designs. Pharmacists and general consumers use cognitive decision-making processes to determine the correct medication, so by optimizing designs with this in mind, the number of mistakes may be able to be reduced greatly.

Ampuero and Vila performed a study on "Consumer Perceptions of Product Packaging" which focused on isolating different aspects of the labels on medical devices such as color, shape, image and typography (Ampuero and Vila, 2006). Isolating individual features on a label helped researchers begin to manipulate and understand how the brain is perceiving and processing the information on the label. Other aspects of the packaging of the medication itself can lead to a specific perception as to what volume is contained within it (Folkes and Matta, 2004). From a commercial manufacturer side, it is important to understand how the consumer views products on a shelf. Young (Young, 2012) presented the "PRS Eye Tracking Method" detailing how products on a shelf are viewed by customers. Understanding how commercial products are viewed and preferred is important specifically in package design, since consumers are generally novices in regards to looking at medication labels.

2. COMMONALITY/DIFFERENTIATION MEASURES

A powerful indicator of the importance of visual information can be its prominence as measured by size and location. To understand how the size and location can connote commonality, Shooter, et al. (2008, 2010) studied nine packages of Tylenol adult acetaminophen. The front panel (the side of the box that faces consumers when placed on store shelves) was analyzed in terms of its features, the text, and the graphics that conveyed information. The front panel was treated as a coordinate and normalized to account for variations in box size. Figure 1 shows the normalized face with each feature element identified and its centroid location. The area of each feature on the normalized package faces was calculated, tabulated, and compared relative to the entire area in order to reflect the package face "real estate" occupied by each feature as shown in Figure 2.

Presumably, features deemed by the manufacturer to be most important take up the most area on the package. The background uses the most space so one might expect consideration of color to be important. Tylenol uses a consistent shade of red for background color across its product family packaging. The brand name Tylenol takes up the second largest area, almost equal to the background. The type of medication and purpose are considerably smaller. Dosage and form are smaller still. The variation in normalized size and location of the Tylenol brand across the nine packages is minimal.

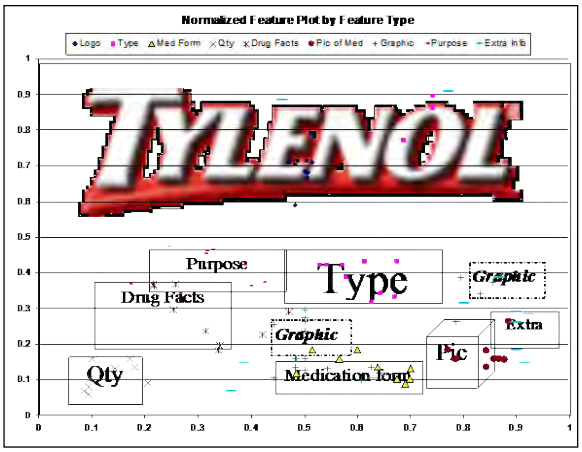


Figure 1: Front Panel Features' Positions

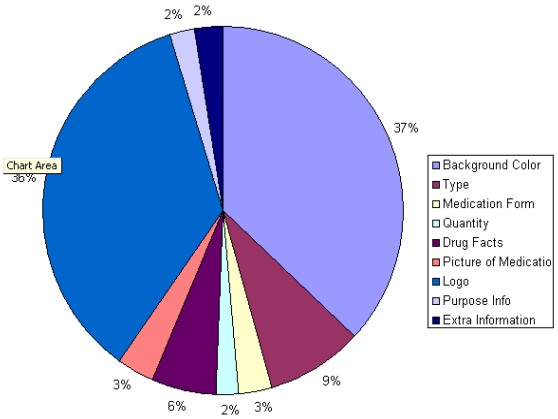


Figure 2: Relative Area of Front Features

There is also considerable consistency in location and size of the "secondary" information such as medication type, purpose, dosage, and form. It is evident that the Tylenol product family utilizes a package platform of features with an emphasis on brand recognition as the most salient common element. Information that differentiates among the variants of the product is less prominent in the package face, but is critical for informed and proper use. It is important to note that the Institute for Safe Medication Practice has recognized Tylenol as having a high number of cases of medication error (Hicks, 2008).

Cohen and Shooter (2010) followed this study with the formulation of measures to represent commonality and differentiation of packaging features with regard to prominence. In this preliminary investigation, the Feature Area Commonality Index (FACI) was formulated for packaging. One advantage of the commonality indices developed by Thevenot et al. (2007) and Alizon, et al. (2009) is that the result ranges between 0 and 1. The progression here is intended to provide a similar scale. The normalized areas for each

feature on each package was determined and tabulated. The mean was then calculated for the area of that feature across the package family. The *Proportion Difference* from the mean is calculated for each instance and represented as shown in Equation 1.

$$Proportion\ Diff = \frac{|\bar{A} - A_j|}{\left(\frac{\bar{A} + A_j}{2}\right)} \quad (1)$$

where $\bar{A}$ is the mean and A_j is the instance value

The measure can help identify and quantify an outlier instance in the packaging family. It is also possible to gain perspective on the total family by calculating the *Average Proportion Difference* for each feature using the value from each package variant. The intent is to describe the degree of commonality for features repeated across variants. If a feature is not present on a package instance, then that instance is not part of the calculation. A value of 1 indicates exact commonality across the variants while 0 means they are different. The *Feature Area Commonality Index (FACI)* is then calculated as seen in Equation 2 below.

$$FACI = 1 - Average\ Proportion\ Difference \quad (2)$$

The FACI provides a measure for each feature (i.e., Brand Logo) across the package family. The FACI for the Brand Logo feature was determined to be 0.92, indicating high commonality. The FACI for the Main Ingredient was 0.68 and for the Medication Use was 0.73, which are also strong indicators of commonality.

It can also be beneficial to represent the aggregate for all of the features. The most direct formulation of the *Aggregate Feature Area Commonality Index (AFACI)* is to take the average of all of the FACI values for the package family. In calculating the AFACI, only the feature categories present in the package family are included. For example, if the package family does not include any instances of the *Flavor Text* feature category, then all instances will have an area of zero, which results in a FACI of 1 (all commonly not present). If these absent features were included in the AFACI, then the result would be skewed toward a higher indication of commonality. This formulation considers all features as equivalent contributors; weighted contributions of different features will be investigated as part of this work. The AFACI for all features of the Tylenol package family studied was calculated to be 0.73, which is a strong indicator of commonality.

A similar approach was taken to formulate the Feature Location Commonality Index (FLCI), which is an indicator of the commonality and differentiation of the location of features across a product family based on clustered distances. We then validated both of these measures through a cognitive workload analysis study with 60 human subjects in Cho et al. (2014). Response time and selection accuracy were found to be positively correlated with the indices.

3. PREVIOUS VALIDATION STUDY

A Penn State University study, *Effects of Over-the-Counter Medication Product Family Design on Knowledge Acquisition and Consumer Preferences* (Cho et al, 2014) sought to study which features in Over the Counter (OTC) medical labeling specifically can be manipulated to encourage consumers to read and process the labels before they choose a medication for purchase. To do so, Robitussin and Equate labels were altered in Adobe Photoshop to create five different variations. These variations included a base design without any emphasized features, a design with increased font size, a label with inclusion of an accent color, a design with an addition of a graphical icon, and a variation with all of the emphasized features. The study was set up as a survey on Qualtrics software (Qualtrics, Provo, UT). Subjects were given symptoms and requested to select the corresponding medication. Accuracy and selection time were measured.

This study found that "variations in labeling and product family design significantly impacts the accuracy and efficiency of medication decision making and thus has the potential to reduce adverse drug events made during the process" (Cho et al, 2014). The study determined that the overall package design did not have a significant impact on the accuracy of a subject's selection. However, increased font size exhibited the shortest selection time, suggesting that increased font increases efficiency. The variation with all of the emphasized features had the longest selection time, which suggests that too many features could be distracting and decrease efficiency in selection. This could point to a limit in how many features can be emphasized before it detracts a feature's prominence and creates clutter. In looking at design recommendations in other avenues of academia, perhaps there is a "design magic number seven" that could be used in label design. Such a concept would limit how many features can be emphasized until they cancel out one another (Miller, 1956).

The study at Penn State also found that a higher commonality of both AFACI and AFLCI resulted in a higher accuracy of selections and a shorter selection time. Packages with this higher AFLCI, or higher commonality of locations of features across the product family, had a lower consumer preference rating from participants. The study concluded by noting that the task was more of a search task than a decision task. It suggested that a future study with an eye tracker or employee tasks more closely related to the decision making process of selecting medications would be appropriate direction to pursue in the future.

Additionally, it was suggested that future studies test a wider range of OTC labels to create a normal distribution for the indices and use product-moment correlation.

This paper investigates and expands upon Cho et al, performed in 2014. It examines areas of particular interest and importance to individuals interacting and giving medications. Within product families, commonality of the specific features of interest were calculated and then compared against accuracy and viewing patterns to further understand how information is processed from medical labels. The study validates the efficacy of the commonality and differentiation measures for visual features on pharmaceutical packaging. Pattern recognition techniques and automated calculation of these measures will support the rapid exploration of alternative designs with the goal of improved dispensing of medications.

4. EXPERIMENTAL PROCEDURE

Participants

Fifty-four undergraduates from a small college in Pennsylvania were used as the participants of this study. Half of the participants were assigned and exposed to the Robotussin and the other half to the Equate stimuli pool. They were part of a psychology class subject pool and received credit for participation.

Materials and procedure

Stimuli used in the Cho study (Cho et al., 2014) were recreated using Adobe InDesign (Viers et al, n.d.) in the same configuration and patterns. The types of labels used included base (with no manipulations), coloring, icon of person, font size and an image with all available manipulations. A total of 50 images were created that each consisted of 4 vertical labels placed side by side with a corresponding question running along the top in the white space. The questions were also recreated and matched from the previous study (Figure 3). The images were then placed into TOBI software (TOBI, Fall Church, VA), 25 Robotussin images and 25 Equate images in their own respective trial setups.

The images that were uploaded into the program were then processed with areas of interest being identified and outlined using the TOBI software (TOBI, Fall Church, VA). These Areas of Interest (AOI) were as follows: brand name, comparison, name, moon/nighttime (specifically used for Robotussin), description, active ingredients, non-drowsy, symptoms, button, and dots/indicators on the body that the drug will apply to. The labels are coded in the following manner: "ImageID#.AOI#.correct/incorrect". The labels were each individually coded as "correct" or "incorrect" relative to the answer to the question to be able to separate the data as such. A sample marked up image can be seen in Figure 4.

Figure 3: Sample stimuli with question of equate, used in the study.

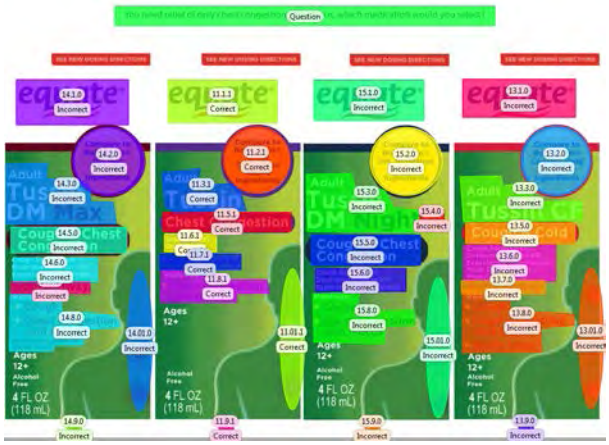


Figure 4: A sample marked up image of a quartet.

Upon entering the study, subjects were given the "Functional Health Literacy in Adults" survey as well as a few general questions about their educational background. This was used to measure how much effort and attention the subject were giving during the study. Afterwards, the TOBI software (TOBI, Fall Church, VA) was opened and the eye-tracking tool was calibrated to the subject. Once completed, subjects were instructed that they were going to be shown images and they would have to read the questions and click on the "correct" image by selecting the circles underneath the corresponding label. Once the subjects had seen all 25 images in their assigned trial, the study was concluded. A sample viewing pattern displayed via heat map can be seen in Figure 5.

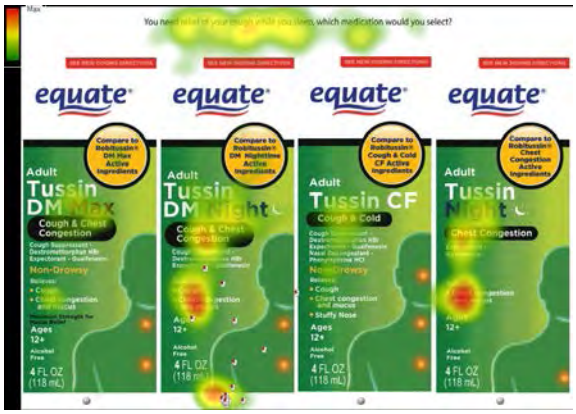


Figure 5: A heat map of one subject's viewing of an image displayed during the study. Yellow indicates shorter viewing times, red displays a longer view time as indicated in the key in the top left corner of the image.

Images were analyzed using the method utilized by (Cho et al, 2014) calculating both location and area commonality of each feature on the label and comparing them within their product families. With this method, product family metrics were used when analyzing the correlation between package design and selection results. Feature Area Commonality Index (FACI) and Feature Location Commonality Index (FLCI) were used to describe commonality of an area of interest. Feature Area Commonality Index (FACI) was determined by calculating the physical area of a label of each specific feature covered.

The FACI was then calculated by subtracting the result from equation 1 from 1, mentioned previously in equation 2. This is useful as if the commonality is consistent across the product family, the FACI is close to 1, and if they are completely differentiated, the result is 0.

Feature Location Commonality Index (FLCI) was determined in a similar manner to the FACI, but the location was determined from the left edge of the package to the centroid of the feature. The locations were again averaged across product families and then subtracted from 1 to calculate the FLCI (Cho et al, 2014). This specific task allowed for analysis of the commonality of location across groups. Each of these values, FACI and FLCI, calculated for feature across product families were then averaged to create aggregates of the FACI and FLCI, further referenced as the AFACI and AFLCI. The AFACI and AFLCI give a good indication as to the commonality across the product family.

Results

An item analysis was conducted, rather than analyzing by subject, so that the effect of commonality on gaze patterns for individual label features (e.g. brand name, active ingredients) could be investigated. For each image, the following were analyzed: the time to first fixation, fixation count, and mean fixation duration for each area of interest (AOI; see Method for description). AOIs that received no fixations were not included in the analysis. The area and location commonality indices (FACI & FLCI, respectively) reported in Cho et al. (2014) for each AOI (see Table 1) were also compared.

A multivariate analysis of variance (MANOVA) was conducted on these five variables across the AOI regions. Results of the MANOVA reveal a significant multivariate effect, Pillai's Trace=1.28, F (30, 1050)=12.08, p<.001, η_p^2=.257, demonstrating a difference in the commonality and pattern of eye movements for the regions of interest. Univariate analyses for this relationship indicate a significant effect of AOI on time to first fixation (F(6,210)=73.36, p<.001, η_p^2=.68), mean fixation duration (F(6,210)=7.78, p<.001, η_p^2=.18), fixation count (F(6,210)=48.33, p<.001, η_p^2=.58), FACI (F(6,210)=15.47, p<.001, η_p^2=.31), and FLCI (F(6,210)=28.81, p<.001, η_p^2=.45).

The primary analysis of interest was the relationship between each feature's commonality (within a product family) and eye-gaze patterns; thus, we conducted a series of bivariate correlations (6 comparisons, Bonferroni corrected α=.008). Both FACI and FLCI were significantly negatively correlated with time to first fixation (FACI: r(216)= -.418, p<.001; FLCI: r(262)= -.535, p<.001; see Figure 1), indicating that for AOIs with greater commonality, participants fixated on that AOI earlier in the trial. In addition, FACI and FLCI were significantly negatively correlated with fixation count (FACI: r(216)= -.354, p<.001; FLCI: r(262)= -.450, p<.001; see Figure 7) and mean fixation duration (FACI: r(216)= -.305, p<.001; FLCI: r(262)= -.193, p=.002; see Figure 8). This suggests that features with greater commonality received fewer fixations and those fixations were shorter in duration.

Table 1. Means and standard deviations of the eye-tracking metrics and commonality indices for the 7 prominent AOIs.

	Time to First Fixation (ms)		Fixation Count		Fixation Duration (s)		AFACI		AFLCI	
Feature/AOI	M	SD	M	SD	M	SD	M	SD	M	SD
Brand Name	29.52	23.64	1.65	1.64	0.31	0.34	0.9793	0.01	0.9907	0.01
Name	94.59	24.98	7.40	3.47	1.13	0.39	0.7067	0.11	0.9391	0.01
Description	102.70	35.66	6.79	2.32	1.00	0.26	0.7987	0.19	0.8856	0.00
Active Ingredients	70.47	37.36	2.87	1.62	0.55	0.41	0.6938	0.26	0.9415	0.04
Non-Drowsy	71.52	19.50	2.28	0.79	0.39	0.15	0.9061	0.12	0.9229	0.01
Symptoms	144.42	32.52	14.73	4.93	1.33	0.45	0.6232	0.32	0.9027	0.06
Dots	16.81	9.88	0.43	0.24	0.10	0.05	0.9686	0.02	0.9681	0.02

Note: Table 1 only displays AOIs that had commonality indices reported in Cho et al. (2014).

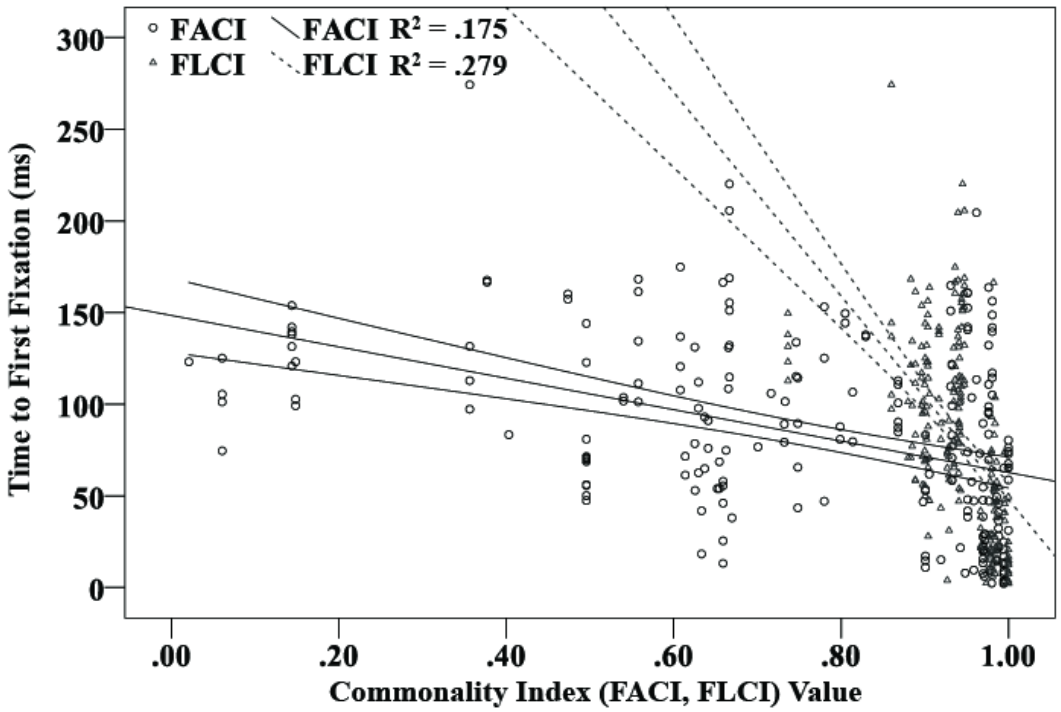


Figure 6. Scatterplot showing relationship between commonality (FACI and FLCI) and time to first fixation (ms).

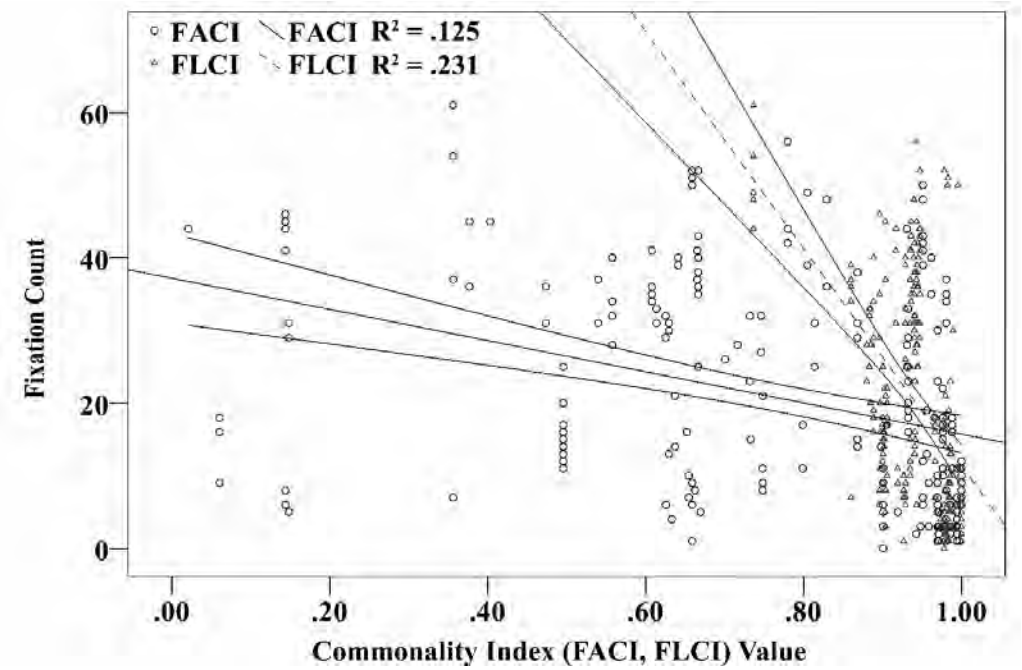


Figure 7. Scatterplot showing relationship between commonality (FACI and FLCI) and mean fixation count across areas of interest (AOIs).

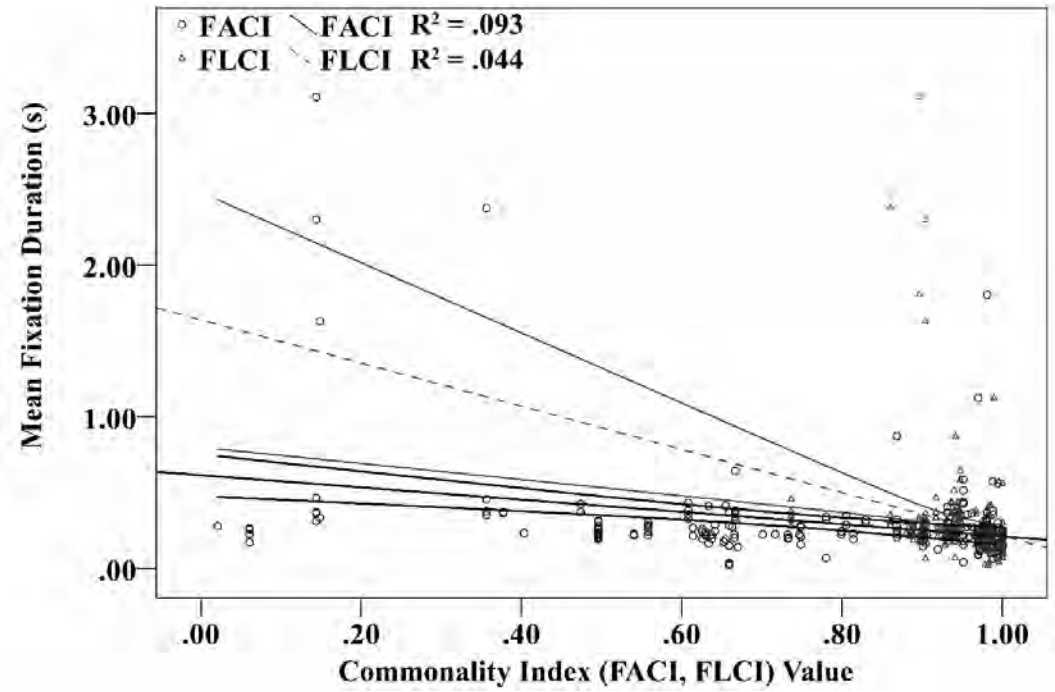


Figure 8. Scatterplot showing relationship between commonality (FACI, FLCI) and mean fixation duration across areas of interest.

5. DISCUSSION AND IMPLICATIONS

Examining the results, it is supported that as subjects become more familiar with the location of certain pieces of information, or specific features on the labels, less time is spent looking at these features. The significance of the correlations we found between the FLCI and FACI reflect this hypothesis. From this data, it is suggested that increasing commonality values among product families, in both location and area, will help to decrease reaction time. An increase in commonality could be applied to both over the counter medication as well as pharmaceuticals.

A shorter reaction time could be helpful and harmful. Decreased reaction time may allow for the pharmacist or consumer to attain more information from the label in a shorter amount of time, or it could decrease the saliency of the information as less time is being spent reading the label. More information in less time would ideally mean fewer mistakes would be made, as the majority of the information on the label would cross through the handler's field of vision. However, more information could also mean an overload of information in the field of vision so less of it is retained. Commonality and regulation among product families affect nurses administering drugs in the hospital setting. For example, if the same information (such as dosage) was in a particular location on every vial the nurse would have to exert minimal effort to determine what the dosage was any given vial, but perhaps they would then minimally view the numerical information given by the feature.

With the knowledge that commonality helps to increase reaction times, future studies will to take this data and build on this concept. A similar study will be performed using prescription labels and medical staff. Since the previous subject pool uses novice medical label viewers, this study would ideally help to support the idea that commonality influence novices as well as experts with years of training. This subject pool is exposed to medical packaging every day, and would be most likely to notice or be affected by changes in labels.

Moving forward new labels will be created, with information that has been manipulated to contain high or low levels of commonalities. The different information found in these areas will test specifically if high levels of commonalities decrease salience of information. It is important to see how the location of information influences the accuracy and consistency of information retrieval as opposed to just reaction time. Results of the current study support that there is less time spent in areas with high commonality, but the researchers are currently unable to determine how salient the information is in those high commonality locations. If commonality decreases reaction time, than perhaps the information provided in these high commonality areas are not as salient as originally intended.

To improve validity of results, comparing two-dimensional renderings of labels and a more realistic rendering is of interest to determine which format will provide a more accurate replica of what nurses would experience in the field handling medical vials. Future studies will investigate the brightness of a computer screen versus brightness of physical labels and how the lighting of testing scenarios might affect the ability for results to extend to a physical pharmacy or hospital dispensary. Studies could also delve into color theory and examine how lumosity might influence choices of medical labels. With a commonality in lumosity or actual color within the feature, perhaps saliency would be able to increase with commonality.

6. IMPLICATIONS FOR DECISION SUPPORT SYSTEM

The study has validated the saliency of the measures for commonality and differentiation of packaging features with improved medication selection. There are tens of thousands of medications on the market both over-the-counter and prescription. The intent is to create a database of diverse pharmaceutical packages that will include the Packaging Commonality Differentiation Indices and highlights of identified "trouble spots". This information will be used to improve upon the current pharmacy dispensary approach where red labels are used as warnings for potential identification hazards, as well as to improve on the internal labeling used for in-patients. The computational models for creating the indices in the database need to be automated as much as possible to relieve burden on entry of new package information. For example, the information capture has been simplified, and the computation of the FACI and FLCI indices has been automated by using software that automatically measures the area and centroid locations of features; and then calculates the indices. However, it is desired to automate the recognition and categorization of features supplied to the measures. Techniques of pattern recognition and cluster analysis will dramatically improve this process. The intent is to develop a system that will provide information to a package designer that will enable the rapid exploration of alterative designs with improved medication administration outcomes.

7. CONCLUSIONS

This paper has introduced a computational model for representing the commonality and differentiation of visual features on pharmaceutical packages. The measures had been previously validated through a

workflow analysis study. This study used eye tracking to evaluate gaze patterns for novice subjects. The results support the measures and indicate that increased commonality of features results in shorter reaction times, but also shorter fixation times. A similar study is currently being conducted with healthcare professionals including nurses and pharmacists to explore the correlations in healthcare settings. The intent is to develop a working decision support system that will support the exploration of alternatives to packaging designers. The implications for decision support in organization is also being examined as well as the structure of pharmaceutical dispensing. The researchers have taken the approach of validating the measures before the development of the decision support system to ensure the efficacy of the approach. The calculation and representation of the measures have been automated. The team is currently exploring techniques for pattern recognition to automate the recognition and categorization of salient features.

8. ACKNOWLEDGEMENTS

This research is supported by the Bucknell Geisinger Research Initiative.

9. REFERENCES

Alizon, F., Shooter, S. B., and Simpson, T. W., 2009, "Assessing and improving commonality and diversity within a product family," Res. Eng. Des., **20**(4), pp. 241–253.

Ampuero, O., and Vila, N., 2006. "Consumer perceptions of product packaging," J. Consum. Mark., 23(2), pp. 100-112.

Aspden, P., Wolcott, J., Bootman, J. L., and Cronenwett, L. R., 2007, Preventing medication errors, National Academies Press Washington.

Cho. J., Miller, S., Simpson, T., and Shooter, S., 2014, "Effects of over-the-counter medication product family design on knowledge and acquisition and consumer preferences," Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), Buffalo, NY.

Folkes, V., and Matta, S., 2004. "The Effect of Packaging Shape on Consumers' Judgements of Product Volume: Attention as a Mental Contaminant," J. Consum. Res., 31(2), pp. 390-401.

Hicks, R. W., Becker, S. C., and Cousins, D. D., 2008, ""MEDMARX data report: report on the relationship of drug names and medication errors in response to the Institute of Medicine's call for action,"" Rockville, MD: Center for the Advancement of Patient Safety, US Pharmacopeia.

Miller, G., 1956. The Magical Number Seven. The Psychological Review.

Qualtrics, (2015). Qualtrics. Provo, UT.

Shooter, S. B., Cohen, S., and Williams, C., 2008, "Assessing Commonality and Differentiation for Packaging Family Planning with Application to Medication Labels," Proceedings of the ASME2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), pp. 67–76.

Shooter, S. B., and Cohen, S. W., 2010, "The Commonality Differentiation Index Using Prominence of Visual Information for Medication Package Family Planning," Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), Montreal, Quebec, Canada, pp. 231–237.

Thevenot, H. J., Alizon, F., Simpson, T. W., and Shooter, S. B., 2007, "An Index-based Method to Manage the Tradeoff between Diversity and Commonality during Product Family Design," Concurr. Eng., **15**(2), pp. 127–139.

Viers, R., Ramos, J., Koren, G., & Brodnitz, D. (n.d.). Adobe InDesign CS5 [Computer software].

Young, S., 2012, "Rigid Plastic: Applying an Architecture," Packag. Des. Mag.

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD-ROM)

Temporal Filtering of Depth Images using Optical Flow

Razmik Avetisyan Christian Rosenke Martin Luboschik Oliver Staadt

Visual Computing Lab, Institute for Computer Science
University of Rostock
18059 Rostock, Germany

{razmik.avetisyan2, christian.rosenke, martin.luboschik, oliver.staadt}@uni-rostock.de

ABSTRACT

We present a novel depth image enhancement approach for RGB-D cameras such as the Kinect. Our approach employs optical flow of color images for refining the quality of corresponding depth images. We track every depth pixel over a sequence of frames in the temporal domain and use valid depth values of the same point for recovering missing and inaccurate information. We conduct experiments on different test datasets and present visually appealing results. Our method significantly reduces the temporal noise level and the flickering artifacts.

Keywords

Temporal Filtering, Optical Flow, Depth Image Enhancement, RGB-D Sensor

1 INTRODUCTION

Today, commodity RGB-D cameras such as the Microsoft Kinect are very popular because of their affordability and the capability to output color and depth images at a high frame rate. They are widely used in computer graphics and virtual reality applications as a low-cost acquisition device.

However, while the color images contain fine details of the scene, the depth images have lower spatial resolution and suffer from extensive noise. The disturbance has a strong temporal component and is perceived as an annoying flickering, even if camera and scene are static. Depth images also contain holes where no depth measurements are available. See Figure 1 to get an impression of the artifacts. Before the depth data can be used in an application, it usually has to be enhanced. There are several existing approaches for this problem that are mostly based on spatial filtering [Chen et al., 2012, Camplani and Salgado, 2012a, Camplani and Salgado, 2012b, Garcia et al., 2013, Yang et al., 2013]. But due to the flickering nature of depth values those approaches oftentimes do not offer satisfactory results.

There are only very few methods that consider the temporal aspect of noise [Matyunin et al., 2011, Islam et al., 2015, Kim et al., 2010]. One reason for this may be the trouble with blurry object boundaries and ghosting artifacts introduced by temporal filtering of dynamic scenes. This happens as temporal filters usually combine depth values of the same pixel from different frames. When the part of the scene represented by that pixel changes over time, which is quite probable in a dynamic scene, then mixing the corresponding depth values is not valid and leads to the mentioned artifacts.

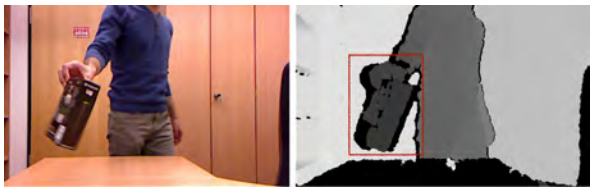


Figure 1: Color and depth images captured using a Kinect RGB-D camera. The depth image contains noise and flickering artifacts while the color image is more robust.

In this work we solve the aforementioned challenges and present a new temporal filtering approach for depth images. We propose to track the movement of objects in the depth image to consistently apply the temporal filter on the same parts of the scene, even if it moves. Based on the detected pixel movements, our method is able to enhance the image quality with standard filtering techniques applied to the temporal domain. However, tracking movements in depth image sequences is a very complicated problem which is even more hampered by their unstable nature, as mentioned above. To circumvent this difficulty, we present a new method for the tracking of movements. As RGB-D cameras simul-

taneously provide color and depth image, we decided to estimate the optical flow of consecutive color images in order to transfer the result to the corresponding depth images. Our idea benefits from the fact that color and depth cameras are usually located close to each other, that is, on the same baseline and with a very small offset. Therefore, we may safely assume that the motion of the imaged scene is induced almost equally on both cameras which eases the transfer.

Having a sound estimation of movement for all consecutive depth frames, we are able to trace back a certain displacement history for each pixel. This provides a one dimensional filtering field for every pixel, which can be processed with any standard 1D filter kernel, such as a simple Gaussian filter. We show that this way largely replaces inaccurate or noisy depth values by valid and stabilized ones. The method can be easily combined with other refinement strategies such as hole filling approaches. We validate our enhancement strategy using two publicly available test datasets and present visually appealing results.

2 RELATED WORK

There are a number of existing approaches that cope with the noise in depth images. Most of them represent classical spatial filtering methods. However, our work is not the first one proposing a temporal approach. A very relevant technique with respect to this article is presented in [Matyunin et al., 2011]. They propose a motion compensation strategy, but only for temporally smoothing depth images. The missing depth pixels are still being recovered from the neighboring pixels, and not from the temporally successive pixels. Furthermore, their approach is an offline approach. An online temporal approach is given in [Islam et al., 2015]. The authors propose to consider the history of depth pixels in the time domain but they do not track the movements. They use a simplified but well parallelizable least median of squares filter to robustly stabilize the depth values. Although their method performs well for static parts of the scene, it exhibits a lot of ghosting artifacts in dynamic parts. In [Kim et al., 2010] the authors propose a combined spatial and temporal depth enhancement method which even applies motion flow between successive color images to infer information about object motion in the corresponding depth images. However, they basically ignore this data in the dynamic parts of the depth images as they use it only to detect stationary parts. Based on this, they apply a bilateral filter to improve the quality which naturally fails in dynamic parts. [Hui and Ngan, 2014] enhance depth images captured from a moving RGB-D system. They also estimate the optical flow of consecutive color images. However, instead of building a temporal filter on top of the obtained data, their method estimates additional depth cues from the flow which are then combined with the original depth images. Their method is intended for mobile setups and cannot be applied to stationary cameras as mainly considered in our case.

Apart from that there are many standard spatial filtering approaches. However, some of them incorporate the information from the color image into the filtering, which relates them to our work. One, proposed in [Camplani and Salgado, 2012a, Camplani and Salgado, 2012b], uses a joint bilateral filter which combines depth and color information. It is working well for static scenes only. The work presented in [Chen et al., 2012] also uses a joint bilateral filter to fill the holes in the depth images. The corresponding color images are used to find and remove wrong depth values near to the edges. Their approach fails to work well for parts where the color image contains a dark region. Other works that incorporate color information for enhancing the quality of the corresponding depth images are presented in [Garcia et al., 2013, Yang et al., 2013]. These approaches provide quite good results in real-time. To sum up, the above mentioned approaches are reducing the noise by using spatial filters mostly. But overall there are few temporal filtering methods that remove the noise caused by moving objects while having stationary cameras.

3 PROPOSED METHOD

To fix the unstable nature of depth pixels captured by RGB-D cameras, we propose a new strategy that enables temporal filtering by keeping track of depth pixels in the time domain. We save a movement history for each depth pixel among a sequence of consecutive frames which is used to validate and correct pixels values. For tracking depth pixels in the time domain, our method employs optical flow [Radford and Burton, 1978], which describes the probable motion of pixels in pairs of consecutive depth frames of a video stream. As the depth stream is too noisy for the accurate estimation of optical flow, we calculate optical flow for the much more stable color video, usually delivered alongside depth data, and apply it for the depth pixels.

In our framework, we assume that an RGB-D sensor continuously provides a sequence of color and depth frame pairs (I_i, D_i). By $I_i(x,y)$ we denote the color of pixel (x,y) in the i-th color frame. Similarly, $D_i(x,y)$ refers to the depth value of pixel (x,y) in the i-th depth frame. While receiving this data in real time, our method always keeps the latest n image pairs. For every frame (I_p, D_p) presently delivered, we use the information in the whole subsequence to produce an improved version D'_{p-m} of the depth image D_{p-m} in the sequence. Hence, every output frame is build on an m-element

preview and an $(n-m-1)$-element history. Clearly, the value of n basically affects memory consumption whereas the value of m influences the latency of our method.

Each incoming pair (I_p, D_p) of frames is firstly inserted at the beginning of our monitored sequence while the oldest one, (I_{p-n}, D_{p-n}), is discarded. Next, we establish two motion fields M_p, N_{p-1} between the new color frame I_p and the previously first color frame $I_p - 1$. While N_{p-1} describes the forward, that is, natural motion of pixels in time, M_p helps to trace back movements. In N_{p-1}, each pixel (x,y) holds a 2D vector (u,v) describing the path taken by the pixel (x,y) from $I_p - 1$ to I_p. More precisely, $N_{p-1}(x,y) = (u,v)$ states that the color value of pixel (x,y) in the image $I_p - 1$ can be traced back to the pixel $(x+u, y+v)$ in the image I_p, that is,

$$I_{p-1}(x,y) \approx I_p(x+u, y+v). \tag{1}$$

While this makes it possible to follow the movement of a pixel along the sequence of frames, it does not help very much to tell where a pixel came from. Hence, here we use M_p where $M_p(x,y) = (u',v')$ states that the color value at $I_p(x,y)$ can be traced back to the previous frame at $I_{p-1}(x+u', y+v')$. As we perform this procedure in every step, we can assume that we have motion fields M_i and N_{i-1} for the pair I_i and I_{i-1} of consecutive color frames for all i in $\{p, ..., p-n+1\}$.

In the next step, we apply the estimated motion fields of the color image sequence to track the history and follow the future of pixels in the corresponding depth images. In particular, for every depth pixel (x,y) in D_{p-m}, we traverse through the available n depth frames following the respective motion vectors. That means, we obtain a sequence $(x_p, y_p), (x_{p-1}, y_{p-1}), \ldots, (x_{p-n+1}, y_{p-n+1})$ of pixel coordinates by setting

$$(x_i, y_i), = \begin{cases} (x,y), & \text{if } i = p-m, \\ (x_{i-1}, y_{i-1}) + & \\ \quad N_{i-1}(x_{i-1}, y_{i-1}), & \text{if } p \le i < p-m, \\ (x_{i+1}, y_{i+1}) + & \\ \quad M_{i+1}(x_{i+1}, y_{i+1}), & \text{if } p-m < i < n. \end{cases}$$

Ideally, this sequence accurately describes the past and prospective motion of the scene object represented at the pixel (x,y) in frame D_{p-m}. That means, we can represent the depth of this object in another sequence $d_p, d_{p-1}, ..., d_{p-n+1}$ of m prospective and $(n-m-1)$ historic depth values by defining $d_{p-i} = D_{p-i}(x_{p-i}, y_{p-i})$ for all $i \in \{0, \ldots, n-1\}$. Figure 2 illustrates this concept.

Recall that the motion fields are derived from the color image. That means for the identified depth sequence that we might get slightly varying depth values due to the z-movement of objects and because of the present noise.

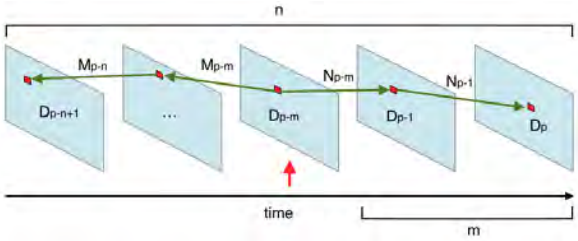


Figure 2: Motion compensated sequence of depth values. For each pixel (x,y), we iterate over a short sequence of prospective and historic depth frames using the motion fields M and N.

Finally, to stabilize the noise in depth image D_{p-m}, we basically filter the n depth values of every pixel, which represents a temporal filtering approach. In our method we apply a weighted filter as follows:

$$D_{p-m}(x,y) = \frac{\sum_{i=0}^{n-1} \omega_i d_{p-i}}{\sum_{i=0}^{n-1} \omega_i} \tag{2}$$

The weights ω_i can be chosen to model certain filter kernels, as for instance a Gaussian filter:

$$\omega_i = e^{-(m-i)^2} \tag{3}$$

Beside static kernels like this, we also support motion dependent kernels, where the weight ω_i is determined by the amount of motion in frame D_{p-i} at pixel (x,y), that is, by the length of the vector $M_{d-i}(x,y)$, respectively of $N_{d-i-1}(x,y)$. This can be used to adaptively reduce the impact of highly dynamic depth frames in which a misinterpretation of real movements is more likely.

4 EXPERIMENTS

For experiments, we fixed the parameters of the method described in Section 3. To keep the latency of our approach low and minimize the ghosting artifacts, we chose to consider a 5-frame history by letting $n = 5$ and we set $m = 0$. This means that there was no preview. Furthermore, to keep the setup simple and to demonstrate our method's potential, we decided to just use a plain averaging filter in the 1D temporal domain. Hence, we set $\omega_i = 1$ for all i. The optical flow was estimated in real time by the method of [Brox et al., 2004] implemented in hardware.

To test the performance of our approach, we have applied different datasets captured with a Kinect camera, each containing at least one moving subject or object. Beside some self-created datasets, we conducted experiments on two publicly available datasets from [Camplani and Salgado, 2014]. Figure 3 demonstrates the method's visually appealing results using our own test sets. Apparently, as seen in the right image, noise and missing depth information, that essentially disturb

the original depth frame in the left image are noticeably fixed or at least reduced by our approach. We like to point out, that the visual improvement covers both, static and dynamic parts of the scene. Furthermore, we also significantly remove flickering, that is, temporal artifacts.

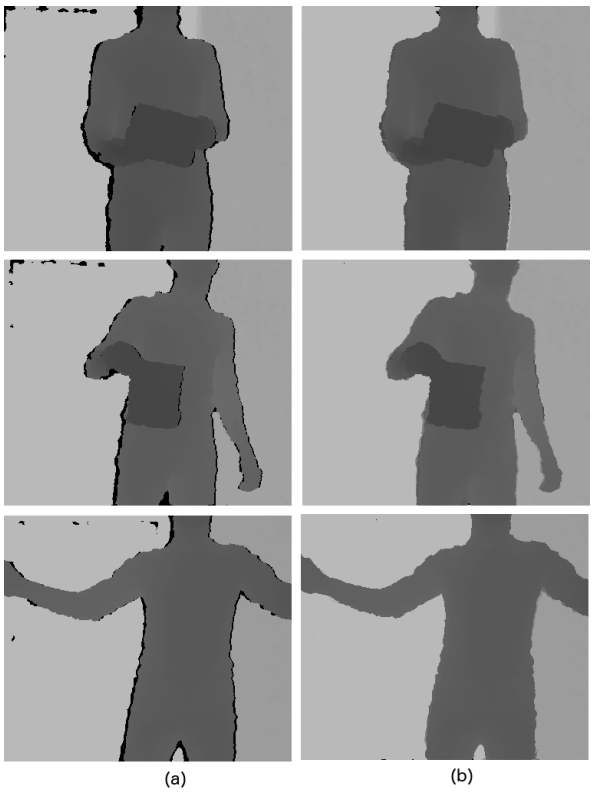


Figure 3: Results for our own datasets. (a) original raw depth images. (b) depth images enhanced by our approach.

Using the datasets from [Camplani and Salgado, 2014], we can also compare the performance of our approach to another state of the art spatial filtering technique for depth image enhancement as described in [Garcia et al., 2013]). For both sets, as depicted in Figure 4, our method fixes most of the missing information and reduces temporal noise for both, static and dynamic parts. Beside that, we get nicer and finer edges around objects.

Limitations of our approach are twofold. Firstly, missing data or noise that stays persistently in one region of the depth image sequence can not be recovered by our temporal filtering approach. In this case, spatial filters, as presented in [Garcia et al., 2013], may perform better. Secondly, artifacts introduced by the motion fields can essentially influence the quality of the output. Even though the color images are more stable and of a higher resolution, it happens that the estimation of optical flow based on the RGB data does not correlate well with the actual movement of objects in the image. Therefore, we sometimes get invalid motion vectors which deteriorate the estimated history of depth values. In particular, we observe that fast movements still cause slight ghosting artifacts, especially for bigger parameter values of n.

Our current implementation runs on the GPU and allows to achieve 10 frames per second. At least in case of average temporal filtering, this speed is basically independent of the choices for the parameters m and n, which only affect memory consumption and latency. For other filter kernels, which do not allow for an incremental update, the performance will also depend on n.

5 CONCLUSION

In this work, we have introduced a new strategy to enhance the quality of depth images using optical flow estimated by the corresponding color images. We have tested our approach with different datasets and presented visually appealing results. It remains future work to fine-tune the method for its full potential by evaluating different parameters m and n and higher order temporal filters. It would also be nice to consider longer histories and even preview to some extend. However, in this case the small errors which build up over time would also have an increased impact. Therefore, to address this problem, we will consider a Gaussian filter which levels the impact of history and preview depending on the temporal distance to the current frame. Aside from that, we plan to combine our new method with other refinement strategies. For instance, we consider to include a spatial filtering into our temporal approach for a more robust enhancement.

6 REFERENCES

[Brox et al., 2004] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV*, chapter High Accuracy Optical Flow Estimation Based on a Theory for Warping, pages 25–36. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Camplani and Salgado, 2012a] Camplani, M. and Salgado, L. (2012a). Adaptive spatio-temporal filter for low-cost camera depth maps. In *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*.

[Camplani and Salgado, 2012b] Camplani, M. and Salgado, L. (2012b). Efficient spatio-temporal hole filling strategy for kinect depth maps. volume 8290, pages 82900E–82900E–10.

[Camplani and Salgado, 2014] Camplani, M. and Salgado, L. (2014). Background foreground segmentation with rgb-d kinect data: An efficient combination of classifiers. *Journal of Visual Communication and Image Representation*, 25(1):122 – 136. Visual Understanding and Applications with RGB-D Cameras.

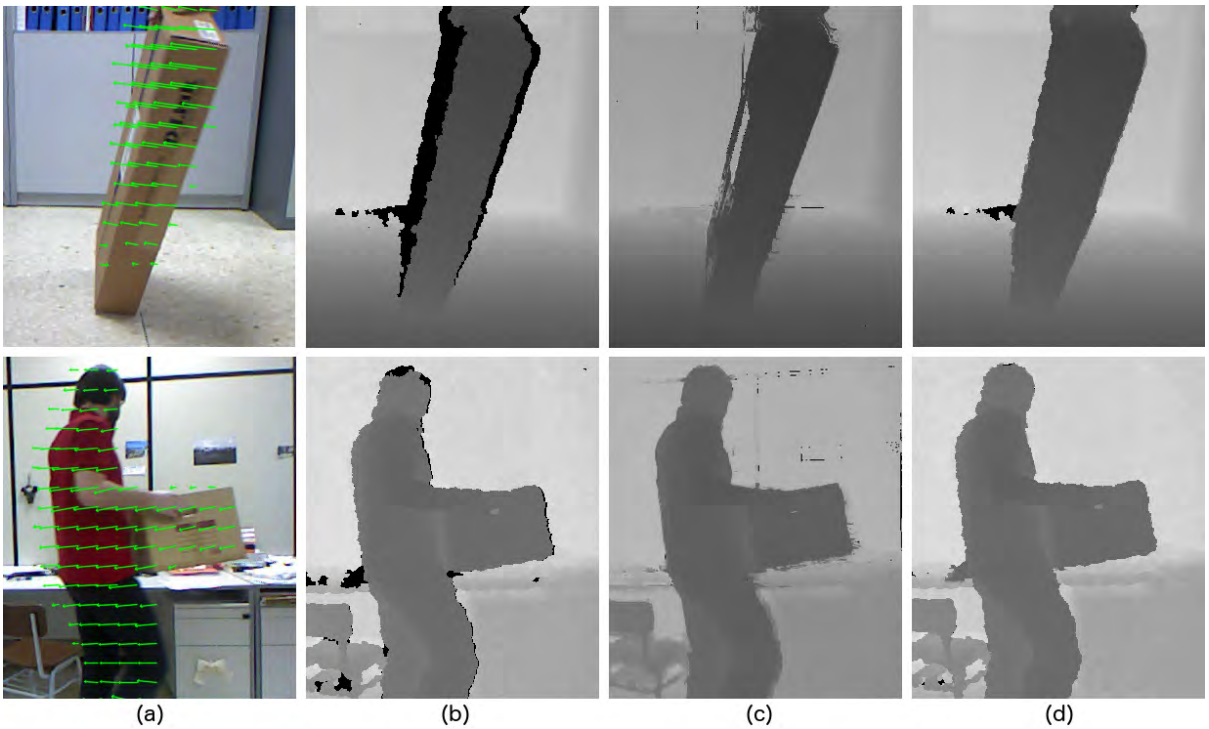


Figure 4: Results for test datasets from [Camplani and Salgado, 2014] - (a) optical flow obtained from the color images, (b) raw depth images (c) output by method from [Garcia et al., 2013] (d) our result. Notably, our results are better for the dynamic parts of the scene.

[Chen et al., 2012] Chen, L., Lin, H., and Li, S. (2012). Depth image enhancement for kinect using region growing and bilateral filter. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3070–3073.

[Garcia et al., 2013] Garcia, F., Aouada, D., Solignac, T., Mirbach, B., and Ottersten, B. (2013). Real-time depth enhancement by fusion for rgb-d cameras. *Computer Vision, IET*, 7(5):1–11.

[Hui and Ngan, 2014] Hui, T.-W. and Ngan, K. N. (2014). Motion-depth: Rgb-d depth map enhancement with motion and depth in complement. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3962–3969.

[Islam et al., 2015] Islam, A. T., Scheel, C., Pajarola, R., and Staadt, O. (2015). Robust enhancement of depth images from kinect sensor. In *Virtual Reality (VR), 2015 IEEE*, pages 197–198.

[Kim et al., 2010] Kim, S.-Y., Cho, J.-H., Koschan, A., and Abidi, M. (2010). Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2358–2361.

[Matyunin et al., 2011] Matyunin, S., Vatolin, D., Berdnikov, Y., and Smirnov, M. (2011). Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4.

[Radford and Burton, 1978] Radford, J. and Burton, A. (1978). *Thinking in perspective : critical essays in the study of thought processes / edited by Andrew Burton and John Radford*. Methuen London.

[Yang et al., 2013] Yang, Q., Ahuja, N., Yang, R., Tan, K.-H., Davis, J., Culbertson, B., Apostolopoulos, J., and Wang, G. (2013). Fusion of median and bilateral filtering for range image upsampling. *Image Processing, IEEE Transactions on*, 22(12):4841–4852.

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD-ROM)

Flexible Calibration of Color and Depth Camera Arrays

Razmik Avetisyan Christian Rosenke Oliver Staadt

Visual Computing Lab, Institute for Computer Science
University of Rostock
18059 Rostock, Germany

{razmik.avetisyan2, christian.rosenke, oliver.staadt}@uni-rostock.de

ABSTRACT

In this work we present a flexible approach for calibrating an array of multiple stationary color and depth cameras using an optical tracking system. Our application domain is focused on 3D telepresence. Calibrating cameras in this area is still a major problem due to the limited applicability of common calibration approaches. Usually, groups of cameras are calibrated relative to each other by either requiring heavily overlapping fields of view for many pairs of participating cameras or free movable cameras.
Our method moves away from these techniques by calibrating every camera individually. The key technology applied is a tracked calibration target with permanently identified global location provided by a tracking device. Detecting the known target geometry in a camera image provides, beside intrinsic calibration parameters, the position of the camera relative to the target. Combining these two aspects of the calibration target's location makes it possible to register every camera in the common tracking coordinate system. We validate our approach using our prototype with 12 Firewire color cameras, 3 Kinect depth cameras, an OptiTrack tracking device, and a checkerboard with an attached trackable rigid body (see Figure 1). In this setup, we achieve a reprojection error of below 0.5 pixels on average.

Keywords

Camera Calibration, Registration, Camera Arrays, Telepresence

1 INTRODUCTION

Multi-camera acquisition setups combining color and depth cameras have become more and more popular in the recent years. A typical example is given by telepresence systems, where the user and the space around her have to be captured from an array of cameras [Maimone and Fuchs, 2011]. A common technical difficulty in the realization of telepresence systems and other multi-camera setups is their accurate calibration, desired at sub-pixel level. Without calibrated cameras, processing the parallelly produced imagery becomes much harder, if not impossible, especially in real time.

The challenge of camera calibration is to find a set of internal and external parameters that describe the physical and geometrical characteristics of all involved cameras and their mutual relations. Intrinsic parameters of a camera, like focal length, principal point, and lens distortion coefficients, describe the non-linear behavior in the projection of scene objects onto the image plane. If known, these parameters can be used to correct the non-linear distortion in the recorded images such that

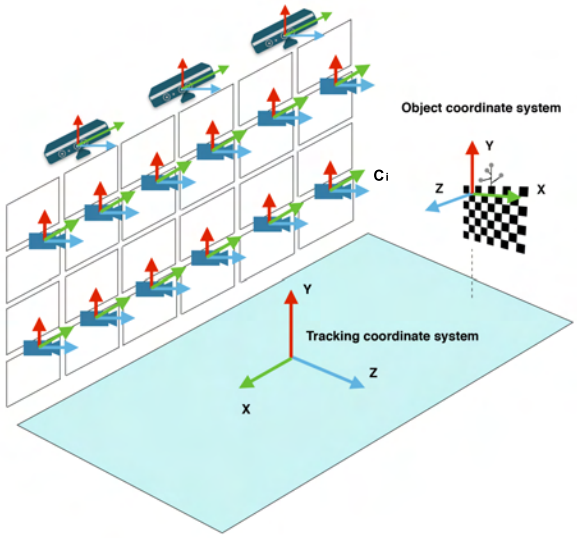


Figure 1: Schematic presentation of our telepresence prototype with 12 RGB and 3 RGB-D cameras integrated into the bezels of a large high-resolution display (LHRD).

they can be treated as if obtained by a pin hole camera. In addition, if the camera is a depth sensor, the intrinsic parameters should also describe the distortion of reported depth values to be able to correct them. The external calibration parameters, on the other hand, describe the camera positions and orientations. Basically, they can be given by a translation vector and a rotation matrix per camera to register them in a joint coordinate system.

Whereas intrinsic calibration of color cameras can be considered as a solved problem in the setting of telepresence, for instance by using a checkerboard and the OpenCV functions based on Zhang's method [Zhang and Zhang, 2000], obtaining extrinsic calibration parameters remains difficult.

Most extrinsic calibration methods available today (see for instance [Bouguet, 2004, Szeliski and Shum, 1997]) assume that many subsets of cameras, usually consisting of two cameras, have an essentially overlapping field of view. The geometrical relation between the cameras of one group is obtained by placing an object of known geometry, like a checkerboard, into the shared field of view. The global setup is subsequently obtained by fitting together the local parameters. However, joining the local parameters coming from different groups is a numerically involved optimization problem. Moreover, the small local errors in every parameter subset tend to add up into a considerable global error. Other extrinsic calibration schemes rely on conditions that are not satisfactory in a telepresence scenario, like free movable cameras, for instance.

To address the problem of calibrating multiple cameras with not necessarily overlapping fields of view, we extend our idea from [Avetisyan et al., 2014]. Similar to our previous work, we apply a tracking system to determine position and orientation of a calibration target in a global coordinate system. Although such tracking systems are quite expensive, they are usually part of telepresence systems to provide a natural interaction experience to users [Lehmann and Staadt, 2013]. In contrast to [Avetisyan et al., 2014], we use this reference data not only to perform a depth correction for cameras, but also to register every camera individually in the global coordinate system. Furthermore, we achieve higher accuracy by using a more robust arrangement of tracking markers. Basically, for every camera individually, the target is placed into the field of view and intrinsic parameters are obtained. Subsequently, the features of the target are detected in a sequence of intrinsically corrected images. Combining the known geometry, position and orientation of the target with the coordinates of the projected features determines position and orientation of the camera with respect to the global tracking coordinate system.

Our prototype setup includes 12 Firewire color cameras with a resolution of 1024×768, arranged in a planar configuration, three depth cameras with a resolution of 640×480 and a 6DOF tracking system (see Figure 1). We use a checkerboard with an attached trackable rigid body. In our setup we were able to accurately calibrate the cameras such that the reprojection error fell below 0.5 pixels on average. The approach described in this work can be easily integrated into a telepresence system, like the one presented in [Willert et al., 2010].

The remainder of this paper is organized as follows: In Section 2, the existing methods are summarized. Section 3 presents the proposed approach for calibrating cameras. In Section 4, we evaluate our approach and, finally, concluding remarks are given in Section 5.

2 RELATED WORK

Most state-of-the-art camera calibration approaches are based on calibration targets with known geometry such as a checkerboard (see for instance [Zhang and Zhang, 2000]) like in our case. Today, there are a number of according toolboxes available [Bouguet, 2004, Barreto et al., 2003, Geiger et al., 2012, Scaramuzza et al., 2006, Svoboda et al., 2005] that implement such an approach for the calibration of setups consisting of two or more cameras. However, most of these methods have the limitation that they make use of overlap in the fields of views for many pairs of participating cameras. Subsequently, we provide a short overview on techniques like these, which are basically distinguished by the specific calibration target that has to be observed in all captured images of a specific camera subgroup. In fact, the simplest possible calibration target geometry is provided by a single light point, which is used in [Barreto et al., 2003, Svoboda et al., 2005]. In [Christoph et al., 2011] the authors show that, for cameras with overlapping fields of view, an active calibration target, like a display showing a temporally varying pattern, can provide better results than a static one. The work [Li et al., 2013], on the other hand, proposes a static pattern that contains much more features than others. In this way only a small fraction of the target has to be visible within each camera image. In [Fernández-Moral et al., 2014] a target is not explicitly required as, instead, they use an overlap of features in the surrounding planar environment, like walls and the ceiling, for instance. If depth cameras are contained in the setup, specialized calibration targets are required. In [Kainz et al., 2012] the authors describe a method for multiple Kinect cameras, where, to obtain extrinsic parameters, they use a target with Kinect-visible markers that are simultaneously detected by a number of these cameras. In [Teng et al., 2014], the authors firstly compute local calibration parameters

to register the color and depth streams of each Kinect camera and then they interpolate these values across the entire captured volume for registering the cameras relative to each other.

However, in general, telepresence systems do not guarantee overlap in the fields of view. Regardless of the used target, this often makes the above standard approach to multiple camera calibration inappropriate for the considered application. Therefore, to compensate the lack of overlap, several calibration methods apply mirrors [Agrawal, 2013, Hesch et al., 2010, Kumar et al., 2008, Lebraly et al., 2010, Sturm and Bonfort, 2006, Takahashi et al., 2012]. If the target cannot be brought into the field of view of a certain camera then a mirror is used to show at least the target's reflection. However, these techniques tend to be inaccurate in large setups like ours. The problem is that the distance between target and camera grows also with the reflection. Thus, the target becomes too small when seen in a mirror. Moreover, methods like these intensify the whole problem as they also have to determine the positions of the mirrors.

For settings, where cameras have insufficient overlap in there fields view, some other approaches rely on the portability of the cameras. For instance, the setup in [Caspi and Irani, 2001] uses the common motion of two closely bonded cameras over time to recover their geometrical relation. Beside the requirement to freely move the two cameras in space, they should also have the same center of projection. A similar approach for rigidly coupled but movable cameras is given in [Esquivel et al., 2007] using structure and motion techniques. Likewise, [Besl and McKay, 1992] proposes to calibrate depth cameras by moving the whole setup around. Here, a geometric iterative closest point method is used to register 3D points obtained from consecutive depth images. Another category of movable camera calibration is to estimate relative motion by odometry, as for instance in [Brookshire and Teller, 2012, Carrera et al., 2011, Heng et al., 2014, Heng et al., 2013, Lébraly et al., 2010, Schneider et al., 2013]. These methods apply to steadily moving cameras rigidly attached to some vehicle. In telepresence setup, however, cameras are rather rigidly connected to the whole setup and cannot be moved at all. Hence, we cannot take these methods into consideration.

A way of solving the problem with not sufficiently overlapping fields of views, which is more relevant to our application in telepresence, is to utilize the motion of objects in the scene [Makris et al., 2004, Micusik, 2011, Pflugfelder and Bischof, 2010, Radke, 2010, Rahimi et al., 2004, Tieu et al., 2005]. The idea is to not only fix the geometry of the target in advance but also the movement of the target. Based on that prior knowledge a camera can determine its own position by recognizing both, location and time of the target in the recorded images. However, all known methods require some calibration parameters to be given in addition. For example, the authors of [Pflugfelder and Bischof, 2010] assume the intrinsic parameters and the rotation of the cameras to be given in advance. Similarly, the authors of [Micusik, 2011] require the gravity vector directions for each camera to be able to estimate extrinsic parameters from target movements.

Another interesting approach, which has been developed for settings as ours, that is, large scale setup where cameras have rather different fields of view, is presented in [Ataer-Cansizoglu et al., 2014]. The authors scan the collaboration space with an external mobile device (SLAM system), like a simple depth camera, to get a 3D model of the acquisition setup. Afterwards they capture the scene with all cameras of the setup and fit the 2D image data onto the 3D capture. Then the 2D/3D correspondences between the stationary camera images and the 3D scan is used to locate every camera. The main problem with this approach is, that it heavily relies on the quality of depth images, which is known to be unstable and often inaccurate. Furthermore, finding point correspondences between a camera image and the entire 3D model is not straightforward and sometimes fails to work well.

The work [Beck et al., 2013] presents a full telepresence system that builds on up to three Kinect cameras per communication side. Although this system does not completely fit to our needs, as we also have color cameras in our setup, their calibration method represents a first starting point for our work. In particular, a milestone for calibration in this article is the presented way to correct depth values of a Kinect by the use of a tracking system. In their approach they determine the spacial relation of a depth camera that is mounted on a motorized platform and the static planar floor. Using this data as ground truth, they build a 3D lookup table that maps reported depth values to the associated positions in physical space. Recently, we simplified this method [Avetisyan et al., 2014] by replacing the complex motorized setup with a simple trackable checkerboard. The target is placed at various positions in physical space to fill the 3D lookup table of corrected depth values. Then, the recorded pairs of reported distance values from the camera and the 3D position of the target are used to interpolate a complete 3D lookup table with corrected depth values. In [Beck and Froehlich, 2015], the methods from [Beck et al., 2013] and [Avetisyan et al., 2014] are combined and refined, especially for the interpolation of corrected depth values.

Another interesting aspect of camera calibration in [Beck et al., 2013], which motivated us to advance their approach, is the method of extrinsic calibration of depth cameras. Like in our case, their technique applies a tracked calibration target to determine the position and orientation of a camera in two steps, firstly by the relation between camera and target and secondly by the known location of the target within the tracking volume. However, their target, a large box-shaped object, is only applicable for depth cameras. Yet, they suffer from problems with detecting the target in the noisy depth image as well as some numerical trouble also caused by the intense noise. In this paper, we present an enhanced, yet much simpler and faster, way to calibrate arbitrary stationary color and depth camera setups that eliminates the problems caused by the needs for overlapping views, mirrors, movable cameras, or complex error-prone numerical computations.

3 PROPOSED METHOD

In our setup we require that the space in front of the cameras is entirely observed by a tracking device, as for instance a Natural Point OptiTrack device like in our case. Moreover, we need a calibration target that fulfills the following requirements: (1) The exact 3D location of all calibration target features can be obtained from the tracked position and orientation of the target. (2) There are target features that are visible in color cameras while others (or even the same) can be seen in infrared images as recorded by depth sensors. Subsection 3.3 proposes a method to create a trackable checkerboard by attaching a rigid body with tracking markers, such that the calibration features, that is, the corner points between black and white squares, are precisely registered in the local coordinate system of the board. Our checkerboard features are clearly visible in both color and depth cameras.

The following presents our calibration procedure together with the underlying basic ideas. To calibrate a (telepresence) system, we move the calibration target in front of each camera j as shown in the Figure 1 and detect the n 2D feature points

$$F_{ij} = \{(x_{ij1}, y_{ij1})^T, \ldots, (x_{ijn}, y_{ijn})^T\} \qquad (1)$$

of the given calibration target in a sequence of camera images taken at times i. In our particular case F_{ij} consists of the $n = 48$ 2D coordinates for projected corner points between black and white squares on our checkerboard. Along with every feature point set F_{ij}, we synchronously record the global coordinates (t_i, r_i) of the checkerboard, where t_i is a translation vector and r_i a rotation matrix specifying location and orientation of the checkerboard in tracking system coordinates, that is, global coordinates. For every depth camera we vary the distance in our movement and also take a longer sequence to later be able to compute the intrinsic parameters according to [Avetisyan et al., 2014].

3.1 Intrinsic Calibration

With the 2D feature points F_{ij}, we are able to compute intrinsic camera parameters for all cameras. As we use a checkerboard in our experimental setting, we use the Open-CV standard method for intrinsic parameters that is based on [Zhang and Zhang, 2000]. Subsequently we can remove any recorded non-linear distortion in our camera images and at the same time, we obtain corrected 2D feature points F'_{ij}.

The intrinsic parameters for the correction of depth values are obtained by applying our approach from [Avetisyan et al., 2014]. Result of this method is a 3D lookup table that maps every triple (x, y, d) of reported depth value d at pixel (x, y) to a corrected depth value d'. We use the corrected feature points F'_{ij} in combination with the global coordinates (t_i, r_i) and the known geometry of the calibration target to correct the reported depth values. In fact, for every time i, a detected feature point (x, y) in depth camera j comes with a recorded depth value d. Simultaneously, it corresponds to a 3D-point p given by the known geometry and the recorded position and orientation (t_i, r_i) of the target at time i. Accordingly, we can define the value in the lookup table at (x, y, d) to be $d' = \|p - (x, y, 0)^T\|$. As we record sufficiently long feature sequence, we have enough defined entries in the look up table to correctly interpolate values for empty spots. See [Avetisyan et al., 2014] for all the details.

3.2 Extrinsic Calibration

The estimation of extrinsic camera parameters is performed for every camera j individually. In contrast to other approaches, we do not use intersecting fields of view, similarities in motion of multiple cameras, or any other shared information. Instead, we apply the known geometry and coordinates of calibration target features and the actually recorded and corrected features F'_{ij} of camera j at the times i. We begin by computing the coordinates of the target relative to camera j for all times i. In other words, we obtain translation vectors T_{ij} and rotation matrices R_{ij} relative to the coordinate system of camera j. For our particular setting, where we use a checkerboard for a target, we again apply the OpenCV method based on [Zhang and Zhang, 2000] to obtain T_{ij} and R_{ij}.

The set of tuples $(t_i, r_i, T_{ij}, R_{ij})$ over all times i and all cameras j contains all information necessary to compute the extrinsic calibration parameters of the whole system. For every i and j we first calculate the position P_{ij} of camera j in the checkerboard coordinate system at time i:

$$P_{ij} = -R_{ij}^T \times T_{ij} \qquad (2)$$

Then we compute for all times i the translation vector τ_{ij} and the rotation matrix ρ_{ij} of every camera j relative to the global coordinate system as follows:

$$\tau_{ij} = r_i \times P_{ij} + t_i \quad (3)$$

$$\rho_{ij} = r_i \times R_{ij} \quad (4)$$

As a result, we obtain for every camera j a sequence of pairs (ρ_{ij}, τ_{ij}), each specifying position and orientation of camera j in the global coordinate system. In a perfect setting, all elements of this sequence would be the same. However, due to unstable environmental conditions, the calibration result varies over time i and it is up to us, to filter out outliers and faulty values. To determine the extrinsic calibration parameters of each camera j, our idea is to select the pair (ρ_{ij}, τ_{ij}) from the given sequence that minimizes the reprojection error, a widely accepted measure for calibration quality.

To determine the reprojection error for a given pair, we use our knowledge about the calibration target geometry together with the information about its location to compute for every time i a virtual representation of the target. In particular, we have a set V of 3D points representing the features of our virtual target. For the checkerboard, that we use in our experimental setup, V consists of the corner points between black and white squares on a 3D model of our board. Next, we move the virtual features to the recorded position of the real target at a time i by transforming every point $p \in V$ into the global coordinate system using

$$q = p \times r_i + t_i. \quad (5)$$

By that we get a set of transformed feature points V_i representing the target in its location with respect to tracking system coordinates at time i. To estimate the quality of a given pair (ρ_{ij}, τ_{ij}), we transform the point set V_i to the respective coordinate system of camera j by evaluating

$$r = \rho_{ij}^{\mathsf{T}} \times (q - \tau_{ij}) \quad (6)$$

for all points $q \in V_i$ to obtain the virtual feature point set V_{ij} in camera coordinates. Next, we use the intrinsics of camera j to project V_{ij} to the image plane of camera j and by that obtain a set

$$V'_{ij} = \{(X_{ij1}, Y_{ij1})^T, \ldots, (X_{ijn}, Y_{ijn})^T\} \quad (7)$$

of reprojected 2D feature points. Finally, we measure the reprojection error at time i and camera j by

$$\delta_{ij} = \sqrt{n^{-1} \sum_{k=1}^{n} (x_{ijk} - X_{ijk})^2 + (y_{ijk} - Y_{ijk})^2}, \quad (8)$$

the square root of the mean squared error. The pair (ρ_{ij}, τ_{ij}) that minimizes δ_{ij} is returned as the extrinsic calibration result for camera j.

Hence, as a result of the whole procedure we get a set $\{(R_1, T_1), (R_2, T_2), \ldots\}$ of pairs (R_j, T_j) specifying for every camera j the location T_j and the orientation R_j with respect to the global tracking system coordinate system. Clearly, this also yields the pairwise relation of all cameras which is usually determined in classical calibration approaches.

3.3 A Trackable Calibration Target for Color and Depth Cameras

For our calibration method, we have to create a trackable target that has features for both color and depth cameras. Instead of using a target that actually has geometric features that can be detected in depth images, like applied for instance in [Beck et al., 2013], we use a simple checkerboard. Then we detect the corners between black and white squares in the infrared image just as for ordinary color images. Solely the fact that the noisy infrared image is improved by a 5×5 median filter stands as a difference to the procedure for color images.

We create a tracked target by attaching a trackable rigid body with a static configuration of tracking markers on the top of a checkerboard similar to the one used in [Avetisyan et al., 2014]. Although this makes it possible to precisely track the position of the rigid body, the exact relation of these locations to the checkerboard features remains vague. To solve this problem, we attach four additional tracking markers onto the corners of the checkers field on the board. For a better understanding see Figure 2. Next, in an initialization step, we

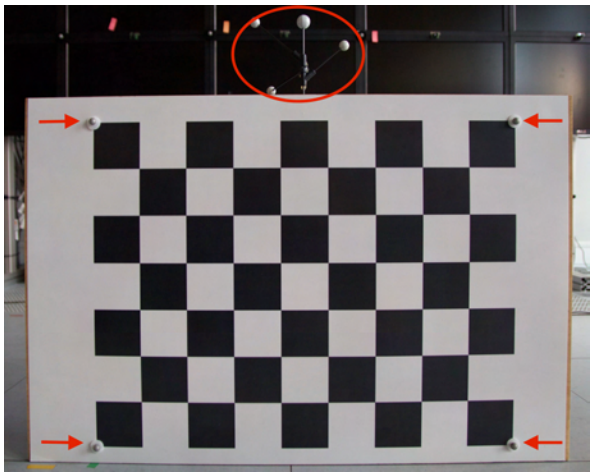


Figure 2: A checkerboard target with attached rigid body and four additional markers on the field corners.

align the rigid body with the checkerboard target making use of the four newly attached markers. For this purpose we first create a virtual marker at the crossing of the diagonals given by the corner markers using the tracking system software. In this way, we exactly get the center of the checkerboard. Then we calculate the

offset from the center to the rigid body's coordinates. Figure 3 illustrates the described procedure. Finally, we

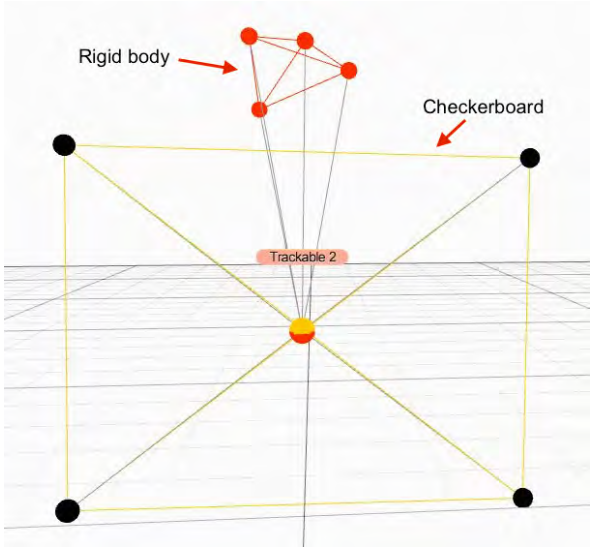


Figure 3: Geometric alignment between the rigid body and the checkerboard.

use the rotation reported from the tracking system and the dimensions of the checkerboard recorded in the initialization to translate coordinates of the rigid body to the left top corner of the checkerboard, which is considered as the origin of our checkerboard coordinate system.

4 EVALUATION

The quality of our calibration method is highly dependent on the accuracy and calibration quality of the used tracking system. For our experiments we used a Natural Point OptiTrack optical tracking system. Twelve infrared cameras surround the calibration space, each of them delivering images with a maximum latency of 10 ms at sub-pixel image accuracy. With our current calibration, the system has around 0.145 mm mean error. We may safely assume that the influence of this error is by magnitudes of order below that of other error sources, like for instance the noise and sampling based inaccuracy in feature detection.

The few linear equations solved in our approach are numerically stable enough to be irrelevant for error estimation. In the following we demonstrate experimentally that other possible error sources have only a very small impact, too.

To evaluate our results quantitatively, we start with the widely accepted reprojection error, which is convenient as it is already implemented and used by our approach. Figure 4 shows an example for error estimation in a arbitrarily selected color image captured by a camera of our setup. For this particular measurement, we observed sub-pixel accuracy of around 0.45 pixels. We

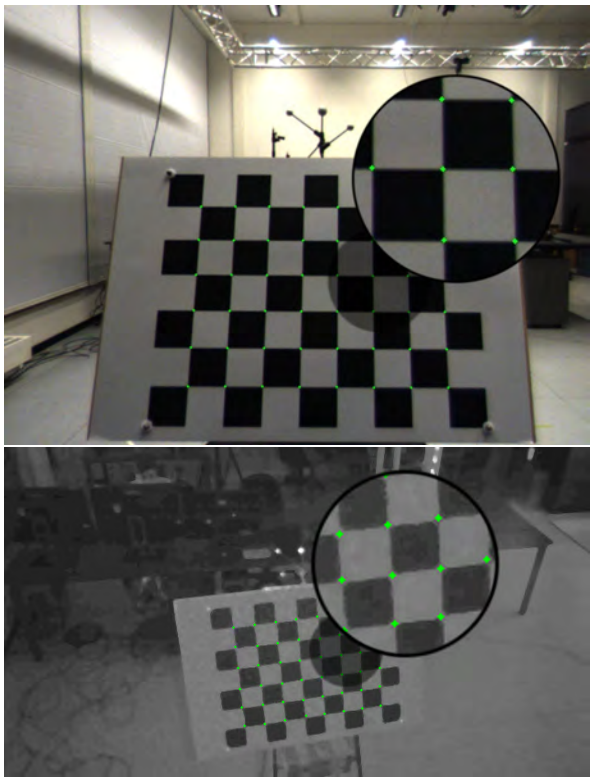


Figure 4: The reprojection error for color images (top) and infrared images (bottom). The reprojected points are shown using green points. Apperently, they are located very closely to the corner points on the checkerboard. See the magnified areas for more details.

also tested different images captured with other cameras and we can report that with our method we are able to achieve reprojection errors permanently smaller than 0.5 pixels. It is worth to mention, that the calibration accuracy varies with the place where the checkerboard was positioned for the capture. In some regions of the tracking volume the tracking system's cameras have a better view on the target and then we observe reprojection errors of up to 0.1 pixels.

The estimation of the reprojection error, as given in this section, states that every camera is accurately registered in the global coordinate system with a similar insignificant error. It follows that also the spatial relation between pairs of cameras is accurately determined as shown in the following experiment where we evaluate the mutual reprojection error between one pair of cameras c_1 and c_2. We place our checkerboard target in the shared field of view of the two cameras and, like in Section 3.1, detect and correct the 2D feature point sets F_1' and F_2' in both camera images. Similar to Section 3.2, using the OpenCV method based on [Zhang and Zhang, 2000] for F_1' provides a translation T_1 and a rotation R_1 specifying the location of the checkerboard relative to camera c_1. Using the spatial relation between c_1 and c_2, which was calculated by

our calibration approach, we can transform (T_1, R_1) to (T_2, R_2) giving the location of the board, as seen by c_1, relative to camera c_2. Then we use the intrinsics of camera c_2 to project the virtual features to the image plane of c_2 and, like in Equation 7, obtain a set V_1' of reprojected 2D feature points. Finally, the reprojection error δ between V_1' and F_2' is obtained as in Equation 8.

Using this method we observed a mutual reprojection error of less than 0.5 pixels. It is worth to mention for our method that, as every camera is individually registered in the global coordinate system, the calibration error in the spatial relation between two cameras does not depend on the actual choice of the cameras or the fact that they have intersecting fields of view. Only for the estimation of the mutual reprojection error, our experiment needed clearly intersecting fields of view.

5 CONCLUSION

We have presented an extremely simple method for reliably calibrating multiple cameras using a tracking system with a trackable calibration target. Our approach does not utilize any further mutual information among neighboring cameras and enables us to calibrate cameras independently from each other in a fast and accurate fashion. Although the proposed method is very flexible and allows to calibrate dense camera arrays, its main weakness comes from the potentially high costs of installing a tracking system in a multi-camera setup, if not yet present.

6 REFERENCES

[Agrawal, 2013] Agrawal, A. K. (2013). Extrinsic camera calibration without a direct view using spherical mirror. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 2368–2375.

[Ataer-Cansizoglu et al., 2014] Ataer-Cansizoglu, E., Taguchi, Y., Ramalingam, S., and Miki, Y. (2014). Calibration of non-overlapping cameras using an external slam system. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 509–516.

[Avetisyan et al., 2014] Avetisyan, R., Willert, M., Ohl, S., and Staadt, O. (2014). Calibration of depth camera arrays. In *Proceedings of SIGRAD 2014, Visual Computing, June 12-13, 2014, Gothenburg, Sweden*, pages 41–48.

[Barreto et al., 2003] Barreto, J. P., Daniilidis, K., Kelshikar, N., Molana, R., and Zabulis, X. (2003). Easycal camera calibration toolbox.

[Beck and Froehlich, 2015] Beck, S. and Froehlich, B. (2015). Volumetric calibration and registration of multiple rgbd-sensors into a joint coordinate system. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 89–96.

[Beck et al., 2013] Beck, S., Kunert, A., Kulik, A., and Froehlich, B. (2013). Immersive group-to-group telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):616–625.

[Besl and McKay, 1992] Besl, P. and McKay, N. D. (1992). A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256.

[Bouguet, 2004] Bouguet, J.-Y. (2004). Camera calibration toolbox for matlab.

[Brookshire and Teller, 2012] Brookshire, J. and Teller, S. J. (2012). Extrinsic calibration from per-sensor egomotion. In *Robotics: Science and Systems VIII, University of Sydney, Sydney, NSW, Australia*.

[Carrera et al., 2011] Carrera, G., Angeli, A., and Davison, A. (2011). Slam-based automatic extrinsic calibration of a multi-camera rig. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2652–2659.

[Caspi and Irani, 2001] Caspi, Y. and Irani, M. (2001). Alignment of non-overlapping sequences. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 76–83 vol.2.

[Christoph et al., 2011] Christoph, S., Frank, F., and Elli, A. (2011). Camera calibration: active versus passive targets. *Optical Engineering*, 50(11).

[Esquivel et al., 2007] Esquivel, S., Woelk, F., and Koch, R. (2007). Calibration of a multi-camera rig from non-overlapping views. In Hamprecht, F., Schnörr, C., and Jähne, B., editors, *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 82–91. Springer Berlin Heidelberg.

[Fernández-Moral et al., 2014] Fernández-Moral, E., Jiménez, J. G., Rives, P., and Arévalo, V. (2014). Extrinsic calibration of a set of range cameras in 5 seconds without pattern. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 429–435.

[Geiger et al., 2012] Geiger, A., Moosmann, F., Car, O., and Schuster, B. (2012). Automatic camera and range sensor calibration using a single shot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3936–3943.

[Heng et al., 2014] Heng, L., Burki, M., Lee, G. H., Furgale, P. T., Siegwart, R., and Pollefeys, M. (2014). Infrastructure-based calibration of a multi-camera rig. In *2014 IEEE International Conference on Robotics and Automation 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 4912–4919.

[Heng et al., 2013] Heng, L., Li, B., and Pollefeys, M. (2013). Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1793–1800. IEEE.

[Hesch et al., 2010] Hesch, J. A., Mourikis, A. I., and Roumeliotis, S. I. (2010). Extrinsic camera calibration using multiple reflections. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 311–325.

[Kainz et al., 2012] Kainz, B., Hauswiesner, S., Reitmayr,

G., Steinberger, M., Grasset, R., Gruber, L., Veas, E., Kalkofen, D., Seichter, H., and Schmalstieg, D. (2012). Omnikinect: Real-time dense volumetric data acquisition and applications. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, VRST '12, pages 25–32, New York, NY, USA. ACM.

[Kumar et al., 2008] Kumar, R., Ilie, A., Frahm, J.-M., and Pollefeys, M. (2008). Simple calibration of non-overlapping cameras with a mirror. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7.

[Lebraly et al., 2010] Lebraly, P., Deymier, C., Ait-Aider, O., Royer, E., and Dhome, M. (2010). Flexible extrinsic calibration of non-overlapping cameras using a planar mirror: Application to vision-based robotics. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5640–5647.

[Lébraly et al., 2010] Lébraly, P., Royer, E., Ait-Aider, O., and Dhome, M. (2010). Calibration of non-overlapping cameras - application to vision-based robotics. In *Proceedings of the British Machine Vision Conference*, pages 10.1–10.12. BMVA Press. doi:10.5244/C.24.10.

[Lehmann and Staadt, 2013] Lehmann, A. and Staadt, O. (2013). Distance-aware bimanual interaction for large high-resolution displays. In Csurka, G., Kraus, M., Laramee, R., Richard, P., and Braz, J., editors, *Computer Vision, Imaging and Computer Graphics. Theory and Application*, volume 359 of *Communications in Computer and Information Science*, pages 97–111. Springer Berlin Heidelberg.

[Li et al., 2013] Li, B., Heng, L., Koser, K., and Pollefeys, M. (2013). A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1301–1307.

[Maimone and Fuchs, 2011] Maimone, A. and Fuchs, H. (2011). Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 137–146.

[Makris et al., 2004] Makris, D., Ellis, T., and Black, J. (2004). Bridging the gaps between cameras. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–205–II–210 Vol.2.

[Micusik, 2011] Micusik, B. (2011). Relative pose problem for non-overlapping surveillance cameras with known gravity vector. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3105–3112.

[Pflugfelder and Bischof, 2010] Pflugfelder, R. and Bischof, H. (2010). Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):709–721.

[Radke, 2010] Radke, R. (2010). A survey of distributed computer vision algorithms. In Nakashima, H., Aghajan, H., and Augusto, J., editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 35–55. Springer US.

[Rahimi et al., 2004] Rahimi, A., Dunagan, B., and Darrell, T. (2004). Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–187–I–194 Vol.1.

[Scaramuzza et al., 2006] Scaramuzza, D., Martinelli, A., and Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5695–5701.

[Schneider et al., 2013] Schneider, S., Luettel, T., and Wuensche, H.-J. (2013). Odometry-based online extrinsic sensor calibration. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1287–1292.

[Sturm and Bonfort, 2006] Sturm, P. and Bonfort, T. (2006). How to compute the pose of an object without a direct view? In Narayanan, P., Nayar, S., and Shum, H.-Y., editors, *Computer Vision – ACCV 2006*, volume 3852 of *Lecture Notes in Computer Science*, pages 21–31. Springer Berlin Heidelberg.

[Svoboda et al., 2005] Svoboda, T., Martinec, D., and Pajdla, T. (2005). A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators and Virtual Environments*, 14(4):407–422.

[Szeliski and Shum, 1997] Szeliski, R. and Shum, H.-Y. (1997). Creating full view panoramic image mosaics and environment maps. In *Computer Graphics (SIGGRAPH'97 Proceedings)*, pages 251–258, Los Angeles. Association for Computing Machinery, Inc.

[Takahashi et al., 2012] Takahashi, K., Nobuhara, S., and Matsuyama, T. (2012). A new mirror-based extrinsic camera calibration using an orthogonality constraint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1051–1058.

[Teng et al., 2014] Teng, D., Bazin, J.-C., Martin, T., Kuster, C., Cai, J., Popa, T., and Gross, M. (2014). Registration of multiple rgbd cameras via local rigid transformations. *IEEE International Conference on Multimedia & Expo*.

[Tieu et al., 2005] Tieu, K., Dalley, G., and Grimson, W. (2005). Inference of non-overlapping camera network topology by measuring statistical dependence. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1842–1849 Vol. 2.

[Willert et al., 2010] Willert, M., Ohl, S., Lehmann, A., and Staadt, O. G. (2010). The extended window metaphor for large high-resolution displays. In *Proceedings of the Joint Virtual Reality Conference of EGVE - EuroVR - VEC, Stuttgart, Germany, 2010.*, pages 69–76.

[Zhang and Zhang, 2000] Zhang, Z. and Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1330–1334.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Measuring Event Probabilities in Uncertain Scalar Datasets using Gaussian Processes

Steven Schlegel
University of Leipzig
schlegel@informatik.uni-leipzig.de

Sebastian Volke
University of Leipzig

Gerik Scheuermann
University of Leipzig

ABSTRACT

In this paper, we show how the concept of Gaussian process regression can be used to determine potential events in scalar data sets. As a showcase, we will investigate climate data sets in order to identify potential extrem weather events by deriving the probabilities of their appearances. The method is implemented directly on the GPU to ensure interactive frame rates and pixel precise visualizations. We will see, that this approach is especially well suited for sparse sampled data because of its reconstruction properties.

Keywords

Gaussian Process Regression, OpenCL Programming, Climate Data

1 INTRODUCTION

The visualization of tensor data that is given on discrete positions typically requires interpolation of the data values in between the sample positions. Usually, this task is solved by using linear interpolation. However, in the case that the given data is uncertain, this method is not feasible [BUR96]. In [SCH12], Schlegel et al. proposed the use of Gaussian process regression to overcome the aforementioned problems.

We will show how this method can be used to calculate occurence probabilities of certain events. Therefore, we present an approach that combines fast computation on the GPU as well as visualizations of arbitrarily dense samplings. This is achieved by using the reconstruction properties of Gaussian Process Regression. We use datasets from the climate research domain to demonstrate our method. One of the major tasks in climate research is the prediction and the understanding of extreme weather events. Events like heat waves, heavy precipitation (resulting in floods) or hurricanes have a large impact on society and politics. Decision makers rely on climate simulation results as accurate as possible. Those results should hold characteristics of extreme events like location, frequency and intensity. There is a lot of research that points out that climate and extreme events undergo a change especially in frequency and intensity. Emanuel [EMA05], for instance, pointed out that the destructiveness of hurricanes immensely increased of the past 30 years. A quick overview for other observed changes can be viewed here: `http://www.ipcc.ch/publications_and_data/ar4/wg2/en/ch10s10-2-3.html#table-10-3`.

2 RELATED WORK

As pointed out by [PAN96] it is important to keep in mind that uncertainty of scalar data can reside in the data value or in the position of the data point or in both. In this paper, we deal with the uncertainty of the data value itself. Pöthkow et al. [POE11a] also targeted this issue and presented an uncertain counterpart for isocontours [LOR87]. Therefore, they calculate the so called level-crossing probability (LCP). In a given interval, the probability is computed that a certain threshold is crossed within. Therefore, they interpolated the expected values and the roots of the central moments to interpolate the probability density function. Schlegel et al. [SCH12] proposed the method of Kriging to interpolate the mean and the variance in an uncertain Gaussian field. They also applied their method to compute LCP. Several acceleration methods were employed in [SCH15] to enable a fast computation of Kriging in 3D scalar fields. They created interactive 3D visualizations of the mean field and showed the confidence of the computation by depicting areas of high uncertainty. Therefore, they computed an upper boundary for the posterior variance. In contrast, we aim to provide probabilities for the data to exceed (or fall below) certain thresholds using the exact posterior variance. [ATH13] analyzed the effects of uncertainty to linear interpolation and isosurface extraction. The extension of [POE11a] to correlated data was done in [POE11b]. To reduce the heavy computation time (mainly caused by Monte Carlo Sampling), two methods called maximum edge crossing probability and linked-pairs to approximate the level-crossing probabilities were introduced in [POE13b]. To overcome the restrictions of predefined probability distributions [POE13a] introduced nonparametric models (empirical distributions, histograms, and kernel density estimators) to compute the probabilty of features in an

uncertain field. Based on [POE11a], Pfaffelmoser et al. [PFA11] developed an algorithm to compute the so called isosurface first crossing probability. It is an algorithm that incrementally uses a front-to-back volume ray casting to visualize that probability. The rendering is enriched by additionally depicting surfaces of the stochastic distance function (SDF-surfaces). Additional work to compute the gradient of the probability density function of uncertain 3D scalar fields was done in [PFA12]. Kniss et al. [KNI05] try to perform classification of medical volume data under uncertainty. They base their transfer function on what they call the decision boundary distance that is computed for every class, which is a maximal log-odds ratio of all the other classes. Roughly speaking, it is a measurement of the risk of being wrong to assume that the current class is the correct one. A more thorough overwiev of visualization of uncertain data can be read in [BRO12].

3 GAUSSIAN PROCESS REGRESSION

As pointed out earlier, we can not rely on standard techniques like linear interpolation, when the need for interpolation of uncertain data arises, . We need to regard the uncertainty of the sampled data points, as well as their respective corellation. If these samples are Gaussian-distributed random variables, it is suitable to use the concept of *Gaussian processes*. Interpolating random variables in a Gaussian process is also known as *Kriging* [KRI51] or *Gaussian process regression* [RAS06]. The basic approach is to assume a prior Gaussian distribution for any (continuous) position in the data set. By considering the given samples and a defined covariance between the data points, this prior distribution is turned into a posterior Gaussian distribution that matches the given data more precisely in the sense of reducing the variance of the distribution. For details, on how to define the prior distribution, we refer to [SCH12].

A Gaussian process given on a domain S defines a Gaussian distributed random variable at any position $s \in S$. It is defined by a mean function at every position s and a covariance function between any two positions s and s', e.g., see [ADL11]:

$$\begin{aligned} \mu : S \mapsto \mathbb{R} \qquad & \mu(s) = \mathbb{E}[f(s)], \\ k : S \times S \to \mathbb{R} \quad & k(s,s') = \mathbb{E}[(f(s)-\mu(s))(f(s')-\mu(s'))], \end{aligned} \tag{1}$$

where the mean function is assumed to be constant. Our choice for the covariance function throughout this paper is the squared exponential. This covariance function models an exponential drop of the covariance with increasing distance of the data points. It is often used in the field of Geostatistics and is given by

$$k(s,s') = \sigma_p^2 exp(-\frac{1}{2l^2}|s-s'|^2), \quad (\sigma_p^2, l > 0). \tag{2}$$

The parameters σ_p^2 (prior variance) and l (length scale) are hyperparameters. Throughout this paper our choice for l will be 1. The choice for σ_p^2 will be discussed in section 4.1. Gaussian processes, as well as the optimization of the hyperparameters, are discussed in detail in [RAS06].

Let S be sampled with N Gaussian distributed variables at positions $s_i,\ i = 1,\ldots,N$, with $X(s_i) = X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and the covariance function $k(s,s')$. Then one can calculate the covariances between those sample points and generate the covariance matrix

$$K = \begin{pmatrix} k(s_1,s_1)+\sigma_1^2 & \ldots & k(s_1,s_N) \\ \ldots & \ldots & \ldots \\ k(s_N,s_1) & \ldots & k(s_N,s_N)+\sigma_n^2 \end{pmatrix}, \tag{3}$$

The posterior distribution at position s is then defined as

$$X(s) \sim \mathcal{N}\left(\vec{k(s)}^T K^{-1}\vec{\mu}_i, \quad k(s,s) - \vec{k(s)}^T K^{-1}\vec{k(s)}\right), \tag{4}$$

with $\vec{\mu}_i$ being the vector of the means of the sampled X_i and $\vec{k(s)} = (k(s,s_1),\ldots,k(s,s_N))^T$. It can be shown that the variance of the posterior distribution is minimized, when estimating $X(s)$ in that way.

Additionaly, by defining the basis functions (see [SCH15])

$$\phi_i(s) = \begin{cases} -1, & \text{if } i = 0 \\ \sum_{j=1}^N (K^{-1})_{ij} k(s_j,s), & \text{otherwise} \end{cases} \tag{5}$$

we can write the computation of $X(s)$ as

$$X(s) = \sum_{i=0}^{N} X_i \phi_i(s). \tag{6}$$

with

$$\begin{aligned} X(s) \sim \mathcal{N}\Big(\mu(s) &= \sum_{i=1}^{N} \phi_i(s)\mu_i, \\ \sigma^2(s) &= k(s,s) - \sum_{i=1}^{N} \phi_i(s) k(s_i,s)\Big). \end{aligned} \tag{7}$$

Using Eq. 6, it is possible to compute the derivative of $X(s)$ by differentiating the basis functions:

$$\begin{aligned} \frac{\delta \mu(s)}{\delta s^{(n)}} &= \mathbb{E}\left(\sum_{i=1}^{N} \frac{\delta \phi_i(s)}{\delta s^{(n)}} X_i\right), \\ \frac{\delta \sigma^2(s)}{\delta s^{(n)}} &= Var\left(\sum_{i=1}^{N} \frac{\delta \phi_i(s)}{\delta s^{(n)}} X_i\right) \end{aligned} \tag{8}$$

This form of Kriging is called *simple* Kriging. Other forms of Kriging differ mainly in the form of the assumption of the prior (e.g. see [GOV97]).

4 METHOD

In this section, we show how to work with Gaussian regression on climate research datasets. Our goal is to calculate the probability that a certain threshold will (not) be exceeded. Thus, we construct uncertainty variables on the basis of the original climate data. We use time dependent data from a global climate simulation given on a rectilinear 2D grid. This could be interpreted as a time series of data at each grid position.

4.1 Modeling the Gaussian Process

First we normalize each time series by removing the seasonal component. Normalized time series have a Gaussian distribution [LAR12]. So the normalization enables us to compute the variability of the data and, of course, to apply our method. Therefore, we replace the value at each time step $v(ts)$ with the average of these values from the annual cycle. For example, if we have monthly means, we replace $v(ts)$ with the average of the period $[ts-6;ts+6]$. Furthermore, we can remove a linear trend in the dataset by replacing each time step with its forward difference, i.e. $v(ts) = v(ts+1)-v(ts)$. Although the trend in temperature time series is not necessarily linear, we can use this simplification for relatively short time periods (i.e. 30 years). This step is optional and best suited for datasets where linear trends may disturb the normalization (e.g. temperature data). In order to estimate the variability of the simulated data at that position, we derive the empirical variance for every grid point based on the normalized time series.

As described in section 3, the basic principle of Gaussian process regression is to turn a prior Gaussian distribution, which we observe on the data, into a posterior Gaussian distribution by taking the given samples and the covariance into account. The prior distribution describes the uncertainty in the data acquisition method. The posterior distribution on the other hand is in general a better estimator for the uncertainty in the dataset in the sense of having less variance. To model the prior distribution, we need it's mean and it's variance. The mean is constantly zero. This can be accomplished by subtracting the empirical mean of the data from the samples. When we display the results, we simply add the posterior mean back on each sample point. This applies to an error model, where the observed value is the sum of the true (unknown) value and a zero mean Gaussian error. The prior distribution variance is the maximum of all the variances which we extracted at the grid points. The maximum variance in the dataset is an obvious choice for the prior variance, because the variance of the data acquisition method is at least as big as the maximum variance residing in the dataset. The prior variance (or signal variance) is the factor σ_p^2 in the covariance function, see eq. 2, which results in $k(s,s) = \sigma_p^2$ (see eq. 4).

Algorithm 1 Creating The Cell Cache on the CPU

```
· l := length scale
· d := cell diameter
· n := number of Cells
· CellCache[n]
for i = 0 to n do
    · b := barycenter(Cell)
    · P := all sample points in radius [b−(3l+d), b+(3l+d)]
    · create and invert covariance matrix using all points in P
    · CellCache[i] := inverted covariance matrix, its positions, and its samples
end for
· send CellCache to graphics card
for all pixels do
    · calculateColor()
end for
```

Algorithm 2 "calculateColor()" – GPU Colormap Algorithm.

```
· t := threshold
· idx := cell index of Pixel
· pos := position in (2D) world space of Pixel
· n := number of sample points in CellCache[idx]
for i = 0 to n do
    · compute basisfunction φ_i using p and CellCache[idx]
end for
· compute distribution using the basisfunctions and Eq. 7
· calculate probability p that the value at pos falls below t
· color pixel according to p and given color map
```

4.2 Implementation

Gaussian process regression performs poorly on many datasets. The reason is the storage and the inversion of the covariance matrix. A method to reduce those requirements, is the use of many small Gaussian processes (and thus covariance matrices) instead of one large process. For regular sampled datasets, it is feasible to create a small Gaussian process for each grid cell composed of the data points lying in a $3l+d$ radius of the bary center of the cell. Where l is the length scale of the covariance function and d is the diameter of the cell. This approach is described in more detail in [SCH12]. The result is, that we have to invert one relatively small covariance matrix for every cell instead of one large matrix, which can also be done in parallel for another speed up.

When an inverted covariance matrix (see Eq. 3) for each cell of the dataset is computed, we send those matrices to the GPU. We also store the dataset itself as well as the indices of the data points that belong

to each of those local Gaussian processes on the GPU. The next step is to compute the probability distribution according to eq. 4 for every pixel position which lies inside our dataset domain. Now, we are able to compute probabilities that values at that pixel fall below or exceed certain thresholds (which in our case are indices for extreme weather events). Furthermore, we are able to accumulate those probabilities over several time steps in order to compute probabilities that the values fall below or exceed the threshold over a given period of time. Given the fact, that we compute everything on the GPU, we finally use the given probabilities for each pixel to create a pixel precise color map which can be rendered immediately by writing the color into the frame buffer. The main advantage of this approach is, that we send the required data (inverted covariance matrix and the point indices) to the GPU once, which will process both tasks, namely computing and rendering. There is no need to send the data back to the CPU. Our implementation uses the OpenCL framework. The algorithm's pseudocode for processing the data on the CPU to send it to the GPU is given in algorithm 1. The pseudocode for the GPU implementation of the calculateColor() function is depicted in algorithm 2.

5 RESULTS

The data we use is a temperature data set from a global climate simulation of IPCC scenario A1B with the coupled atmosphere ocean general circulation model (AOGCM) ECHAM5-MPIOM, which was carried out as a contribution to the International Panel on Climate Change Assessment Report 4 (IPCC AR4) [SOL07]. It is given on a 192 x 96 rectilinear grid. At each grid point, there is a temperature data time series of monthly means from the year 1860 to the year 2100. We used the data from 1860 to 1890 in order to calculate the variances for every grid position.

After the variances are calculated, we assigned them to temperature data (simulated by the same model) given on a 6 hour basis to calculate the probability that the temperature of 273.15°K (0°C) was not exceeded in the whole month of January 2001 (124 time steps). The prior distribution is calculated as described in section 4.1. The result is given in Fig. 1. Additionally, we zoomed into one area containing probability transitions to demonstrate that this kind of interpolation in fact enables rendering using arbitrary zoom factors and still providing smooth results.

This kind of application is also interesting with respect to regional climate changes. Therefore, we used as a second example a data set from a simulation with the regional climate model CLM [HOL08]. This is a community model for the German climate research, originally based on the LM forecast model of the German Weather Service (DWD). The CLM simulation was

Figure 1: Colormap of the probability that the temperature of 273.15°K (0°C) is not exceeded in the whole month of January 2001.

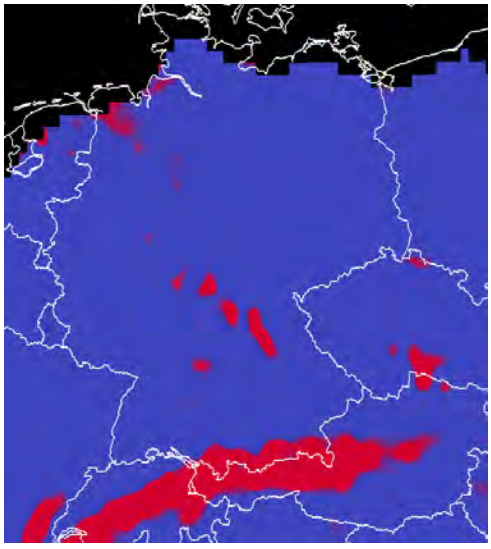


Figure 2: Colormap of the probability that the surface runoff exceeds a threshold of 60 kg/m^2 at least five consecutive days in the summer of 1961.

forced with results of the IPCC scenario A1B simulation with ECHAM5 / MPI-OM. The particular dataset we used is the surface runoff. The surface runoff is the amount of water that cannot be absorbed by the soil. It is an accumulated quantity mainly based on precipitation, snow melting, and the water content of the soil surface and is an indicator for floods. If large volumes of surface runoff flow into a river in a short period of time, the likeliness of a flooding increases. In climate research, one typically counts how often the data exceeds or falls below a threshold within a certain interval to identify weather extremes; see [SIL] and references. As in the example above, we can normalize the time series data at each grid point and calculate the probability that a certain threshold of the surface runoff is exceeded. With our method, we are able to interpolate the data, incorporate the uncertainty of the simulations into the interpolation and compute the probability pixel by pixel. A second step would be to assign the results to the corresponding river catchment basin and accumulate the probabilities over this area to derive potential risks for people living near those rivers, see [SCH13]. Unfortunately this is beyond the scope of this paper.

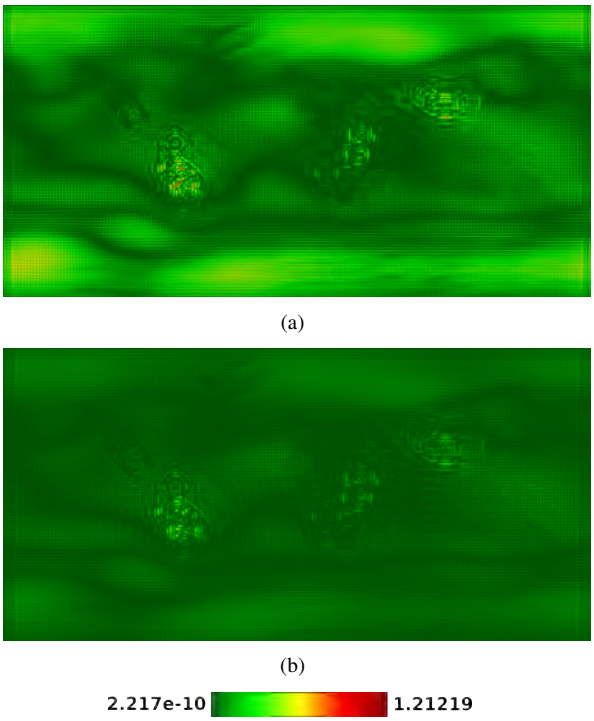


Figure 3: Images showing the regression error for $k = 2$ (Fig.3(a)) and $k = 3$ (Fig.3(b)).

The data we used is a cutout of 65×50 grid points of daily CLM data for Europe centered on Germany. The grid is regular (data stream 3) and has a spacing of 0.2° (approximately 20km). It is a simulation run for the 20th century (20C) from 1961-1990. In Fig. 2, we depicted the probability that the surface runoff exceeds the amount of 60 kg/m^2 in at least five consecutive days in the summer (June, July, and August) of 1961 as a showcase. We can judge from the image that within Germany especially regions inside the catchment basin of the river Rhine have a high probability of exceeding the threshold. This method can be a valuable tool when performing research on larger time scales to evaluate the development of such quantities in order to draw conclusions on climate change.

5.1 Error Analysis

In Section 4.2, we showed that cutting off the exponential covariance function provides smaller cell caches and thus a faster computation of the regression result. On the other hand, this technique introduces errors, which we will analyze using a sample 2D climate data set (sea level pressure) on a 192x96 grid. Therefore, we first calculate the inverted covariance matrix for all the grid points, i.e. we do not cut off the covariance function. Then, we do an regression of the samples with this covariance matrix and use this as a ground truth. The interpolated field again is regulary sampled at 2880x1440 positions. In the next step, the field is interpolated and resampled the same way but using shortened covariance functions, each with a different length: $kl + d,\ k = 1,..,10$. Finally, we compare those fields with the ground truth and calculate the absolute average error.

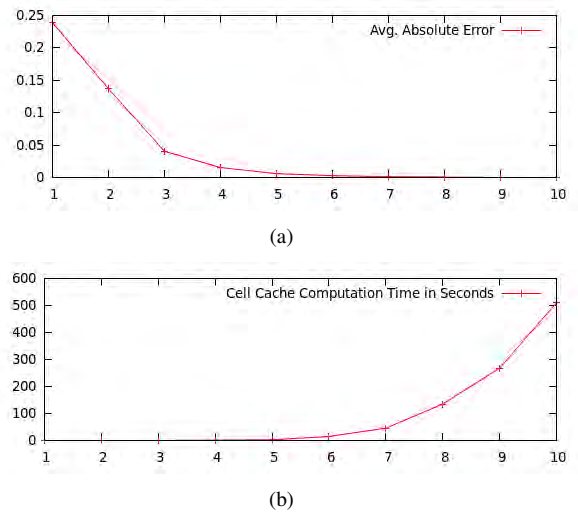


Figure 4: Diagrams showing the exponential drop of the error with increasing covariance influence radius (Fig.4(a)), as well as the exponential gain of computation time (Fig. 4(b)) for different k.

An error colormap for $k = 2$ and $k = 3$ can be seen in Fig. 3. We can conclude from those images, that the error in fact decreases, when the covariance function gains a larger influence radius. But this comes at a comparably high computational cost. The development of error and computational cost to create the cell cache is depicted in Fig. 4. We can conclude an exponential drop of the error as well as an exponential increasing computation time with increasing k. The average relative error for this particular field ranges from $0,0057\%$ to $4,27\%$.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we showed how climate data can be interpolated in an arbitrarily dense matter. Therefore, we use the framework presented in [SCH12] as a basis and extend it by implementing it on the GPU. This enables interactive visualizations with arbitrarily dense samplings. We normalized the time series data to extract the simulation variance. Then we were able to calculate probabilities for certain occurrences like the appearance of extreme events while considering the variance. The resulting color maps can be computed in any desired resolution. Especially when the underlying data is sparse (like in Fig. 2), we nevertheless are able to provide visualizations with smooth transitions.

We want to point out that the variance we used in this paper is computed from the given data. Of course there

are cases, when the variance may known a priori, for example when the simulation model has a known error. The proposed method also works with that kind of uncertainty as long as it is Gaussian distributed.

In general, Gaussian process regression works on any type of scalar data regardless of the underlying topological structure. The only precondition is, that there has to be covariance defined between any of the data points in the given domain. This covariance is often modeled with a covariance function, which then serves as the interpolation kernel. Since we use Gaussian process regression for the means of data interpolation, it is suitable to use a distance based covariance function. With these prerequisites, Gaussian process regression resembles inverse distance based interpolation methods (e.g. Shepard interpolation).

A suitable application of this paper is to study the probabilities of extreme weather events. As mentioned before in Sec. 5, this work can therefore be extended by assigning the calculated probabilities to certain areas of interest to draw conclusions on the danger for flooding, droughts et cetera. We consider this as future work.

7 REFERENCES

[ADL11] Adler, R.J. and Taylor, J.E. Topological complexity of smooth random functions: École d'Été de Probabilités de Saint-Flour XXXIX - 2009. Lecture Notes in Mathematics. Springer, 2011.

[ATH13] Athawale, T. and Entezari, A. Uncertainty quantification in linear interpolation for isosurface extraction. IEEE Transactions on Visualization and Computer Graphics, 19(12):2723–2732, Dec 2013.

[BRO12] Brodlie, K. and Osorio, R.A. and Lopes, A. A review of uncertainty in data visualization. In Expanding the Frontiers of Visual Analytics and Visualization, pages 81–109. Springer, 2012.

[BUR96] Bursal, F.H. On interpolating between probability distributions. Applied Mathematics and Computation, 77(2-3):213 – 244, 1996.

[EMA05] Emanuel, K. Increasing destructiveness of tropical cyclones over the past 30 years. Nature, 436(7051):686–688, 2005.

[GOV97] Goovaerts, P. Geostatics for Natural Resources Evaluation. Applied geostatistics series. Oxford University Press, 1997.

[HOL08] Hollweg, H.-D., Böhm, U., Fast, I., Hennemuth, B., Keuler, K., Keup-Thiel, E., Lautenschlager, M., Legutke, S., Radtke, K., Rockel, B., Schubert, M., Will, A., Woldt, M., and Wunram, C. Ensemble simulations over europe with the regional climate model clm forced with ipcc ar4 global scenarios. Technical Report, 2008.

[SOL07] Intergovernmental. Climate Change 2007 - The Physical Science Basis: Working Group I Contribution to the Fourth Assessment Report of the IPCC. Cambridge University Press, Cambridge, UK and New York, NY, USA, September 2007.

[KNI05] Kniss, J.M., Van Uitert, R., Stephens, A., Li, G. S., Tasdizen, T., and Hansen, C. Statistically quantitative volume visualization. pages 287–294, October 2005.

[KRI51] Krige, D G . A statistical approach to some basic mine valuation problems on the witwatersrand. Journal of the Chemical Metallurgical and Mining Society of South Africa, 52(6):119–139, 1951.

[LAR12] Larsen,R.J., and Marx, M. An introduction to mathematical statistics and its applications; 5th ed. Prentice Hall, Boston, MA, 2012. The book can be consulted by contacting: IT-ES-DNG: Abler, Daniel.

[LOR87] Lorensen, W.E., and Cline, H.E. Marching cubes: A high resolution 3d surface construction algorithm. COMPUTER GRAPHICS, 21(4):163–169, 1987.

[PAN96] Pang, A.T., Wittenbrink, C.M., and Lodh, S.K. Approaches to uncertainty visualization. The Visual Computer, 13:370–390, 1996.

[PFA11] Pfaffelmoser, T., Reitinger, M., and Westermann, R. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. In Computer Graphics Forum, volume 30, pages 951–960. Wiley Online Library, 2011.

[PFA12] Pfaffelmoser, T., Mihai, M., and Westermann, R. Probability distributions for gradient orientations in uncertain 3d scalar fields. Technical report, Technische Universität München, 2012.

[POE11a] Pöthkow, K., and Hege, H-C. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. IEEE Transactions on Visualization and Computer Graphics, 17(10):1393 – 1406, 2011.

[POE11b] Pöthkow, K., Weber, B., and Hege, H.-C. Probabilistic marching cubes. Computer Graphics Forum, 30(3):931 – 940, 2011.

[POE13a] Pöthkow, K., and Hege, H.-C. Nonparametric models for uncertainty visualization. Computer Graphics Forum, 32(3):131 – 140, 2013.

[POE13b] Pöthkow, K., Petz, C., and Hege, H.-C. Approximate level-crossing probabilities for interactive visualization of uncertain isocontours. International Journal for Uncertainty Quantification, 3(2):101 – 117, 2013.

[RAS06] Rasmussen, C.E., and Williams, C. Gaus-

sian Processes for Machine Learning. MIT Press, 2006.

[SCH12] Schlegel, S, Korn, N., and Scheuermann, G. On the interpolation of data with normally distributed uncertainty for visualization. IEEE Transactions on Visualization and Computer Graphics, 18(12):2305–2314, 2012.

[SCH13] Schlegel, S, Böttinger, M., Hlawitschka, M, and Scheuermann, G. Determining and visualizing potential sources of floods. In EuroVis Workshop on Visualisation in Environmental Sciences, Leipzig, 2013.

[SCH15] Schlegel, S., Goldau, M., and Scheuermann, G. Interactive GPU-based visualization of scalar data with gaussian distributed uncertainty. In David Bommes, Tobias Ritschel, and Thomas Schultz, editors, VMV 2015 - Vision, Modeling and Visualization. Eurographics Association, 2015.

[SIL] Sillmann, J., and Roeckner, E. Indices for extreme events in projections of anthropogenic climate change. Climatic Change, 86(1-2):83–104, 2008.

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD-ROM)

Kinematical Ruled Surfaces based on Interrelated Movements in Triads of Contacted Axoids

Galina S. Rachkovskaya
RSTU
1006 Moorefield Hill Place
22180, Vienna, VA, USA
g.rachkovskaya@gmail.com

Yuriy N. Kharabayev
RSTU
1Б Budennovskiy Ave. # 46
344002, Rostov-on-Don, Russia
kharabayev@aaanet.ru

Natalya S. Rachkovskaya
NMF
1006 Moorefield Hill Place
22180, Vienna, VA, USA
narachkovska@gmail.com

ABSTRACT

Kinematical ruled surfaces are constructed by generating the line's motion of a moving ruled surface during its movement along a fixed ruled surface [Spr02a]. The main condition of constructing kinematical ruled surfaces is that a moving axoid contacts with a fixed axoid along their common generating line in each of their positions during the movement of one axoid along another. A lot of well-known kinematical ruled surfaces are constructed on the base of certain pairs of contacted axoids such as “plane – cylinder”, “plane – cone”, “cylinder – cylinder”, “cone – cone”, etc. [Kri06a]. A new model of constructing kinematical ruled surfaces based on interrelated movements in the triads of contacted axoids is proposed in this research. Geometrical models, analytical representations, and computer visualization of the new constructed kinematical surfaces for some cases of triads of contacted axoids “plane – cylinder – cylinder”, “plane – cone – cone”, “cylinder – cylinder – cylinder”, “cone – cone – cone” (Fig. 1), and for matched triads of one-sheet hyperboloids of revolution are developed in this paper. Figures of the triads of contacted axoids and corresponding constructed kinematical ruled surfaces have been developed with the help of the software application AMG (“ArtMathGraph”) [Con07a].

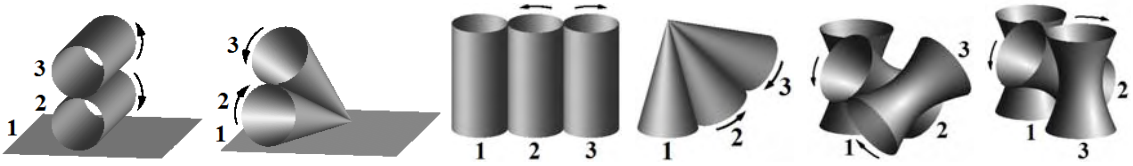


Figure 1

Keywords

Geometrical Modeling, Computer Graphics, Kinematical Surfaces.

1. INTRODUCTION

Kinematical ruled surfaces as a result of one generating line’s motion of the moving ruled surface during its movement along the fixed ruled surface in the cases of certain pairs of contacted axoids are well-known ruled surfaces [Kri06a]. New abilities for constructing kinematical ruled surfaces are originated on the base of the model of interrelated movements in the triads of the contacted axoids, where one of them is a fixed axoid (**1**) and two other are moving axoids (**2**, **3**). The substance of this model consists in the correspondence between the movement of axoid **3** along axoid **2** and the movement of axoid **2** along fixed axoid **1**. The movement of axoid **2** along fixed axoid **1** is accompanied by the backward motion of axoid **3** along axoid **2**, so that the positional relationship of axoids **1, 2, 3** during interrelated movements in the triads is fixed. Examples of the application of the proposed model for constructing rotational ruled surfaces are realized on the base of triads: “plane – cylinder – cylinder”, “plane – cone – cone”, “cylinder – cylinder – cylinder”, “cone – cone – cone” (Part 2.1-2.4). Examples of constructed kinematical ruled surfaces on the base of interrelated movements in the triads of one-sheet hyperboloids of revolution are also realized (Part 3).

2. ROTATIONAL RULED SURFACES BASED ON MODELS OF TRIADS OF CONTACTED AXOIDS

2.1 Model of triad "plane – circular cylinder – circular cylinder"

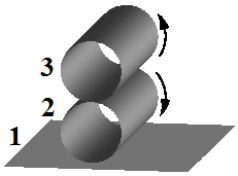


Figure 2

The triad "plane – circular cylinder – circular cylinder" is shown in Fig. 2. In this system of contacted axoids the movement of cylinder **2** along fixed plane **1** is accompanied by the backward motion of cylinder **3** along cylinder **2**, so the axis of moving cylinder **3** is located in the common plane with both axis of cylinder **2** and common generating line of cylinder **2** and plane **1** right along during interrelated movements in this triad of axoids.

Geometrical model of constructing a kinematical ruled surface, generated by one of generating lines of moving cylinder **3**, is presented as a superposition of interrelated movements: rotational movement of moving cylinder **3** around its axis and translational movement of the axis of cylinder **3** along plane **1**. As a result of successive transformations of coordinates, parametric representation (in parameters u, v) of the rotational ruled surface, generated by one of generating lines of moving cylinder **3** in the fixed coordinate system oxyz, connected with fixed axoid **1** (the common generating line of cylinder **2** and plane **1** is lying in axis ox) is:

$x = v$;
$y = u + b\cos\varphi$;
$z = 2a + b(1+\sin\varphi)$,
where $\varphi = u / b$,
a – radius of moving cylinder **2**,
b – radius of moving cylinder **3**.

As it ensues from these parametric equations, the form of constructed kinematical ruled surface is a-independent. In other words, the kinematical ruled surface constructed on the base of triad of axoids "plane – circular cylinder – circular cylinder" (Fig. 3) is the same as the kinematical ruled surface, constructed on the base of the pair of contacted axoids "plane – circular cylinder" [Kri06a].

Figure 3

2.2 Model of triad "plane – circular cone – circular cone"

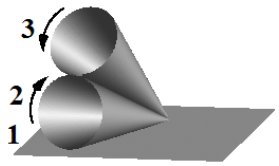


Figure 4

The triad of contacted axoids "plane – circular cone – circular cone" is shown in Fig. 4. The axoid **1** in this triad is a fixed axoid. By perfect analogy with the triad "plane – circular cylinder – circular cylinder", described above (Part 2.1), the movement of cone **2** along fixed plane **1** is accompanied by the backward motion of cone **3** along cone **2** so that the axis of the moving cone **3** is located in the common plane with both axis of cone **2** and common generating line of cone **2** and plane **1** right along during interrelated movements in this triad of contacted axoids (Fig. 4).

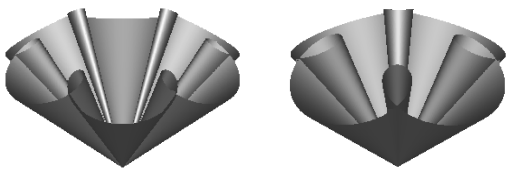
As a result of successive transformations of coordinates, parametric equations (in parameters *u*, *v*) of the rotational ruled surface, generated by one of the generating lines of moving circular cone **3** in the fixed coordinate system *oxyz*, connected with fixed axoid **1**, are defined. The origin of coordinate system *oxyz* is located at the vertex of cone **2** (cone **3**).

Parametric equations of rotational ruled surface are:

$x = X\cos u - (Y\sin\theta + Z\cos\theta)\sin u$;
$y = X\sin u + (Y\sin\theta + Z\cos\theta)\cos u$;
$z = -Y\cos\theta + Z\sin\theta$,
where
$X = v\sin\alpha_3\cos\varphi$;
$Y = v\sin\alpha_3\sin\varphi$;
$Z = v\cos\alpha_3$;
$\theta = 2\alpha_2 + \alpha_3$; $\varphi = (1/\sin\alpha_3)u$

(α_2, α_3 – angles between the cone's generating line and cone's axis for circular cones **2**, **3** accordingly).

Examples of the visualization of rotational ruled surfaces constructed on the base of the triad "plane – circular cone – circular cone" are shown in Fig. 5 (cone **2** ($2\alpha_2 = 40°$), cone **3** ($2\alpha_3 = 20°, 30°$).



Cone **2** ($2\alpha_2 = 40°$), cone **3** ($2\alpha_3 = 20°$) | Cone **2** ($2\alpha_2 = 40°$), cone **3** ($2\alpha_3 = 30°$)

Figure 5

2.3 Model of triad "circular cylinder – circular cylinder – circular cylinder"

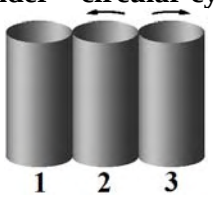


Figure 6

The triad of contacted circular cylinders is shown in Fig. 6. In this system of contacted circular cylinders, the outside surface of moving cylinder **2** revolves around the outside surface of fixed cylinder **1**. At the same time the outside surface of moving cylinder **3** revolves around the outside surface of moving cylinder **2** so the axis of moving cylinder **3** is located in the common plane with both the axis of cylinder **2** and fixed cylinder **1** right along during interrelated movements in this triad of contacted cylinders.

Geometrical model of constructing a kinematical ruled surface, generated by one of the generating lines of moving cylinder **3**, is presented as a superposition of interrelated movements: rotational movement of moving cylinder **3** around its axis and rotational movement of the axis of cylinder **3** around the axis of fixed cylinder **1** lying in axis oz of the fixed coordinate system oxyz, connected with fixed axoid **1**.

Parametric equations (in parameters u, v) of the kinematical ruled surface, generated by one of the generating lines of moving cylinder **3** in the fixed coordinate system oxyz are:

$x = c\cos\varphi\cos u - (R + c\sin\varphi)\sin u$;

$y = c\cos\varphi\sin u + (R + c\sin\varphi)\cos u$;

$z = v$,

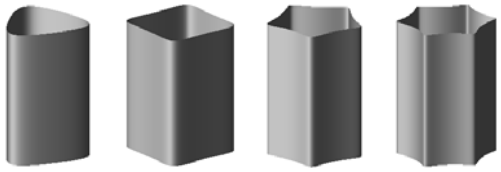
where $R = a + 2b + c$, $\varphi = -(a/c)u$,

a – radius of fixed cylinder **1**,

b – radius of moving cylinder **2**,

c – radius of moving cylinder **3**.

Examples of the computer visualization of rotational ruled surfaces constructed on the base of the triad of contacted circular cylinders are shown in Fig. 7 (ratio of contacted cylinder's radii, i.e. ratio $a:b:c$).



(3:1:1) (4:1:1) (5:1:1) (6:1:1)

Figure 7

2.4 Model of triad "circular cone – circular cone – circular cone"

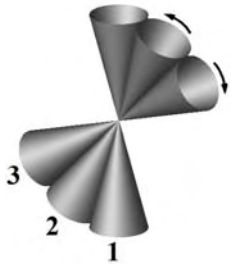


Figure 8

In the triad of contacted circular cones (Fig. 8) cone **1** is a fixed axoid. The fixed cone's axis is lying in the axis *oz* of the fixed coordinate system *oxyz*, connected with fixed axoid **1** (the origin of the coordinate system *oxyz* is located in the vertex of fixed cone **1**). By perfect analogy with the triad of contacted circular cylinders described above (Part 2.3), the movement of cone **2** along fixed cone **1** is accompanied by the backward motion of cone **3** along cone **2** so that the axis of moving cone **3** is located in the common plane with both axis of cone **2** and axis of cone **1** right along during interrelated movements in this triad of contacted cones (Fig. 8).

Parametric equations (in parameters *u*, *v*) of the rotational ruled surface, generated by one of the generating lines of moving cone **3** in the fixed coordinate system *oxyz,* are:

$x = X\cos u - (Y\cos\theta - Z\sin\theta)\sin u$;

$y = X\sin u + (Y\cos\theta - Z\sin\theta)\cos u$;

$z = Y\sin\theta + Z\cos\theta$,

where

$X = v\sin\alpha_3\cos\varphi$;

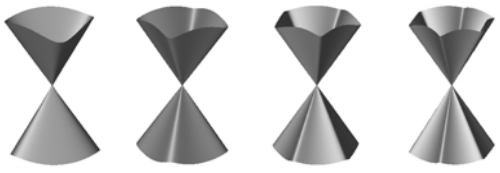
$Y = v\sin\alpha_3\sin\varphi$;

$Z = v\cos\alpha_3$;

$\theta = \alpha_1 + 2\alpha_2 + \alpha_3$; $\varphi = -(\sin\alpha_1/\sin\alpha_3)u$

(α_1, α_2, α_3 – angles between the cone's generating line and cone's axis for cones **1**, **2**, **3** accordingly).

Examples of the computer visualization of rotational ruled surfaces constructed on the base of the triad of contacted circular cones are shown in Fig. 9 ($\sin\alpha_1 : \sin\alpha_2 : \sin\alpha_3$ - ratio of cone's parameters).



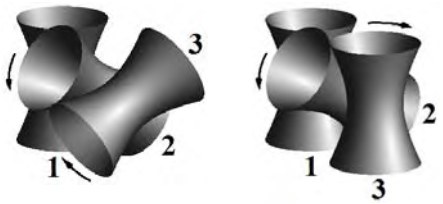
(3:1:1) (4:1:1) (5:1:1) (6:1:1)

Figure 9

3. KINEMATIC RULED SURFACES BASED ON TRIADS OF ONE-SHEET HYPERBOLOIDS OF REVOLUTION

3.1 Geometrical models of triads of one-sheet hyperboloids of revolution

Two possible variants of positional relationship of contacted axoids **1, 2, 3** in the matched triads of one-sheet hyperboloids of revolution are shown in Fig. 10. One-sheet hyperboloid of revolution **1** is a fixed axoid in both configuration variants of triads of contacted axoids.



Variant 1 Variant 2

Figure 10

In correspondence with the proposed model of interrelated movements in the triad of contacted axoids, as the base of constructing kinematical ruled surfaces, the movement of axoid **2** along fixed axoid **1** is accompanied by the backward motion of axoid **3** along axoid **2** (as it is shown in Fig. 10), so the positional relationship of contacted one-sheet hyperboloids of revolution **1, 2, 3** is fixed during interrelated movements in this triad of axoids. The case when interrelated movements in this triad of contacted axoids are realized, so that the center of the waist circle of moving axoid **3** is located in the common line with centers of waist circles of both moving axoid **2** and fixed axoid **1** right along during interrelated movements in this triad, has been described in this research. It is necessary to notice here that the main condition of constructing kinematical ruled surfaces based on the pairs of contacted axoids (Fig. 11) is that the moving axoid contacts with the fixed axoid along their common generating line in each of their positions during the movement of one axoid along another.

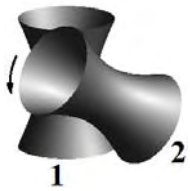


Figure 11

In the cases described above (Parts 2.1–2.4) such moving as rolling one axoid along another is sufficient to meet this main condition. At the same time such moving as rolling one axoid along another in the case of one-sheet hyperboloid of revolution as fixed and moving axoids is insufficient to meet the main condition of constructing kinematical surfaces. However, as it follows from the earlier research [Con09a], the task of constructing kinematical ruled surfaces moves in this case to feasible solution on the base of *complex moving* one axoid along another. *Complex moving* is a combination of several concerted movements of one axoid along another.

In the case of the pair of one-sheet hyperboloids of revolution (as fixed and moving axoids), the geometrical model of *complex moving* one axoid along another as the base of constructing kinematical ruled surfaces can be represented as a superposition of three interrelated movements [Kri15a]:

(1) rotational movement of the moving axoid around its axis;

(2) rotational movement of the moving axoid's axis around the fixed axoid's axis;

(3) translational movement of the moving axoid along the common generating line of both axoids.

Besides, as it was determined in the earlier research [Con09a], for fulfillment of the main condition of constructing kinematical ruled surfaces based on the *complex moving* one axoid along another in the case of the pair of different contacted one-sheet hyperboloids of revolution, the parametric condition

$$a_1^2 + c_1^2 = a_2^2 + c_2^2$$

for the matched pair of contacted axoids must be in progress.

Parameters a_1, c_1 and a_2, c_2 are parameters of the canonical equation of the matched pair of fixed (**1**) and moving (**2**) axoids accordingly.

(The canonical equation of the one-sheet hyperboloid of revolution [Kor61a]:

$\frac{x^2}{a^2} + \frac{y^2}{a^2} - \frac{z^2}{c^2} = 1$, where a – radius of waist circle).

Consequently, in the case of interrelated movements in the triad of contacted one-sheet hyperboloids of revolution (Fig. 10) as the base of constructing kinematical ruled surfaces, *complex moving* of axoid **2** along fixed axoid **1** must be accompanied by the backward *complex moving* of axoid **3** along axoid **2** as it is shown in Fig. 10.

In addition, for triads of different contacted one-sheet hyperboloids of revolution the parametric condition

$$a_1^2 + c_1^2 = a_2^2 + c_2^2 = a_3^2 + c_3^2$$

for matched triads of contacted axoids must be in progress (a_1 c_1, a_2, c_2 and a_3, c_3 are parameters of the canonical equation of matched triad of fixed (**1**) and moving (**2, 3**) axoids accordingly).

3.2 Analytical representation and computer visualization of new constructed kinematical ruled surfaces

In the geometrical model of the triad of contacted one-sheet hyperboloids of revolution (Fig. 10) the axis of fixed axoid **1** is lying in the axis oz of the fixed coordinate system oxyz, connected with fixed axoid **1** (the origin of the coordinate system oxyz is located in the center of the waist circle of fixed one-sheet hyperboloid of revolution **1**).

As a result of successive transformations of coordinates, the parametric equations (in parameters u, v) of a new kinematical ruled surface, generated by one of the generating lines of moving axoid **3** in the fixed coordinate system oxyz are:

$x = (X\cos\theta + Z\sin\theta)\cos u - (a_1 + 2a_2 + a_3 + Y)\sin u$;

$y = (X\cos\theta + Z\sin\theta)\sin u + (a_1 + 2a_2 + a_3 + Y)\cos u$;

$z = -X\sin\theta + Z\cos\theta$, where

$X = -a_3\sin\varphi + a_3 v\cos\varphi$;

$Y = a_3\cos\varphi + a_3 v\sin\varphi$;

$Z = c_3 v$;

$\varphi = -(a_1 / a_3)u$;

$\theta = \theta_1 + 2\theta_2 + \theta_3$ (Variant 1 in the Fig. 10),

$\theta = \theta_1 - \theta_3$ (Variant 2 in the Fig. 10);

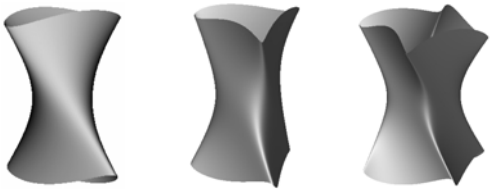
$\theta_1 = arctg(a_1 / c_1)$; $\theta_2 = arctg(a_2 / c_2)$;

$\theta_3 = arctg(a_3 / c_3)$.

Examples of the computer visualization of kinematical ruled surfaces, constructed on the base of both variant 1 and variant 2 (Fig. 10) of triad's configurations of contacted one-sheet hyperboloids of revolution, are shown in Fig. 12 (ratio of waist circles radius of axoids **1, 2, 3** as $a_1 : a_2 : a_3$).

(2:1:1) (3:1:1) (4:1:1)

Variant 1 of triad's configuration



(2:1:1) (3:1:1) (4:1:1)

Variant 2 of triad's configuration

Figure 12

Computer representation of figures for triads of contacted axoids and computer construction of new kinematical ruled surfaces has been realized by the previously developed software application AMG ("ArtMathGraph") [Con07a].

4. CONCLUSIONS

Thus, the new geometrical model for constructing kinematical ruled surfaces based on interrelated movements in triads of contacted axoids is developed in this research. On the base of this model, the analytical representation and computer visualization of new constructed kinematical ruled surfaces is realized for some cases of triads, so as "plane – cone – cone", "cylinder – cylinder – cylinder", "cone – cone – cone", and the matched triad of one-sheet hyperboloids of revolution. The new proposed geometrical model in the combination with the graphic ability of the previously developed software application gives improved opportunity for computer search of desirable kinematical ruled surfaces.

5. REFERENCES

[Spr02a] Sprott, K., Ravani, B. Kinematic generation of ruled surfaces. Advanced in Computational Mathematics, 17: 115-133, 2002.

[Kri06a] Krivoshapko, S.N., Ivanov, V.N. Khalabi, V.N. Analytical Surfaces. Nauka, Moscow, Russia, 2006, 544 p.

[Con07a] Rachkovskaya, G.S., Kharabayev, Yu.N., and Rachkovskaya, N.S. Computer composition of the transformed classical surfaces as the ways and means of the construction of visual models of realistic objects (The new software application "ArtMathGraph") Proceedings of the 15-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2007, Plzen, Czech Republic, p.p. 29-32.

[Con09a] Rachkovskaya, G.S., Kharabayev, Yu.N. Geometric modeling and computer graphics of kinematic ruled surfaces on the base of complex moving one axoid along another (one-sheet hyperboloid of revolution as fixed and moving axoids). Proceedings of the 17-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2009, Plzen, Czech Republic, p.p.31-34.

[Kri15a] Krivoshapko, S.N., Ivanov, V.N. Encyclopedia of Analytical Surfaces. Switzerland, Springer, 2015, 752 p.

[Kor61a] Korn, G.A. and Korn, T.M. Mathematical handbook for scientists and engineers. McGraw-Hill, NY, USA, 1961, 720 p.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

MorphableUI: A Hypergraph-Based Approach to Distributed Multimodal Interaction for Rapid Prototyping and Changing Environments

Andrey Krekhov
High Performance Computing
University of Duisburg-Essen
47057, Duisburg, Germany
andrey.krekhov@uni-due.de

Jürgen Grüninger
Intel VCI
Saarland University
66123, Saarbrücken, Germany
juergen.grueninger@dfki.de

Kevin Baum
Intel VCI
Saarland University
66123, Saarbrücken, Germany
baum@intel-vci.uni-saarland.de

David McCann
Intel VCI
Saarland University
66123, Saarbrücken, Germany
mccann@intel-vci.uni-saarland.de

Jens Krüger
High Performance Computing
University of Duisburg-Essen
47057, Duisburg, Germany
jens.krueger@uni-due.de

ABSTRACT

Nowadays, users interact with applications in constantly changing environments. The plethora of I/O modalities is beneficial for a wide range of application areas such as virtual reality, cloud-based software, or scientific visualization. These areas require interfaces based not only on the traditional mouse and keyboard but also on gestures, speech, or highly-specialized and environment-dependent equipment.
We introduce a hypergraph-based interaction model and its implementation as a distributed system, called MorphableUI. Its primary focus is to deliver a user- and developer-friendly way to establish dynamic connections between applications and interaction devices. We present an easy-to-use API for developers and a mobile frontend for users to set up their preferred interfaces.
During runtime, MorphableUI transports interaction data between devices and applications. As one of the novelties, the system supports I/O transfer functions by automatically splitting, merging, and casting inputs from different modalities. MorphableUI emphasizes rapid prototyping and, e.g., facilitates the execution of user studies due to easy UI reconfiguration and device exchangeability.

Keywords

Dynamic interfaces; scenario-dependent interaction; rapid prototyping.

1 INTRODUCTION

Present-day technologies allow applications to run in heterogeneous and changing environments. Different environments provide users with different input and output devices. Even in the same environment, users typically have different needs and preferences with respect to such interaction devices. This wanted flexibility creates a demand for user interfaces that are adaptable to changing environments and user preferences by spanning the plethora of contemporary I/O modalities and devices. However, the engineering workload involved in making applications fully adaptable in this sense is very high, and, as a result, applications nowadays often support only a limited number of devices.

Consider the following use-case: A group of experts wants to perform a deep brain stimulation on a patient. This kind of brain surgery requires various medical datasets to be explored in advance as well as being monitored during the process. Further assume that a 3D visualization application that is able to handle those datasets is available. In the preparation stage, the experts review the dataset at the office. The interaction setup involves well-known devices such as mouse and keyboard, and the dataset is displayed on a monitor. Later, the experts meet in the conference room and review the surgery roadmap on a large display wall while standing in front of it. The interaction is done via gestures, speech, and personal mobile devices. During the surgery, the doctor relies on a big touchscreen to monitor the process and change parameters on the fly via touch-based or Leap-Motion-captured gestures. The latter is a benefit in aseptic environments where touching should be avoided or is not possible.

The scenario above outlines three different environments and workflows based on the same application but with different interaction requirements. One way to tackle this issue is to add support for various devices to the

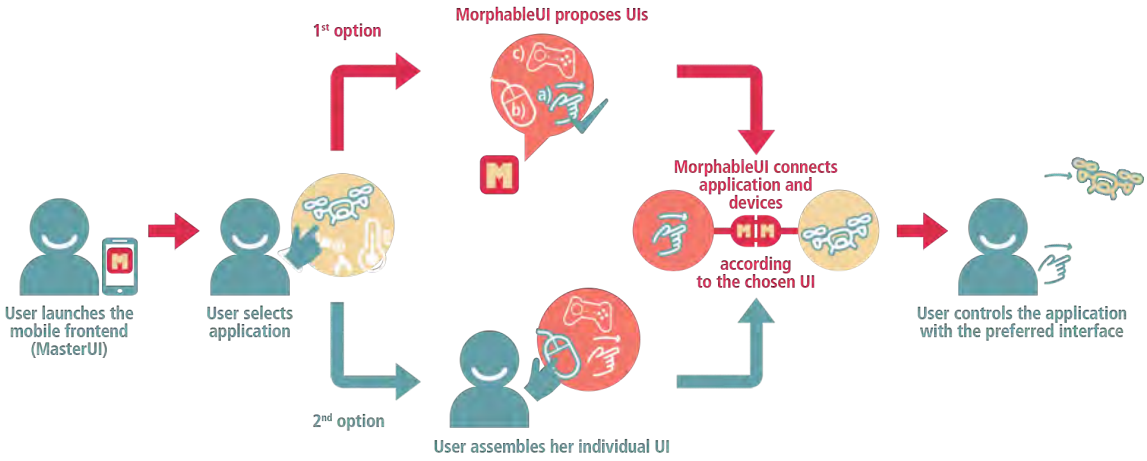


Figure 1: MorphableUI guides users through the UI configuration process by proposing UIs learned from previous decisions and assists with manual UI configuration. During runtime, the system dynamically connects the associated application, devices and interaction data streams over a network.

application itself and extend that range when needed. When new types of devices are introduced within the field of an application, the latter must be modified in order to accommodate these new interaction possibilities. In contrast to this approach, we propose a model that allows for dynamically connecting applications and devices in a way that makes user interfaces adaptable to changing environments and user preferences. This method helps developers avoid having to adapt their application to various types of devices and enhances rapid prototyping possibilities. Users are given the freedom to control applications in the way that best suits their needs by making use of any device that is available in their environment.

In addition, we present an implementation of the proposed model. The implementation provides a uniform, easy-to-use API for arbitrary devices and applications and exposes a service that allows users and developers to configure and dynamically change their interfaces. We demonstrate the service's capabilities by a mobile application suitable for rapid and easy reconfiguration of user environments.

2 RELATED WORK

To solve the outlined issues, two major tasks need to be addressed. First, a way to abstract from actual devices, manufacturers, and even modalities to cover all available interaction possibilities is required. Future devices should also be captured by the developed abstraction. Second, one needs a way to dynamically set up and modify interfaces by taking into account the environment and user preferences. A number of different approaches, especially to the first task, have been presented in the past. Our work builds on these achievements and establishes an interaction environment that includes device and application classification, UI generation, and I/O data streaming.

2.1 I/O Abstraction

I/O hardware abstraction layers hide the details of the underlying hardware. They are often used in VR/AR/MR environments where one has to deal with various kinds of often highly specialized I/O equipment such as motion tracking or 6 degree-of-freedom (6 DOF) devices. One example of the latter is the Control Action Table (*CAT*) [13]. It combines both 3D and 2D interaction techniques and extends the UI design space. Another option is to combine the CAT with other devices such as HMDs or the sensors of a smartphone. In the case of 6 DOF controls, one should also pay attention to the human ability to coordinate movements [27]. One well-established approach to wiring input devices and applications that is used in a number of VR environments is the *VRPN* [25] system. Apart from introducing abstract classes such as joysticks, VRPN streams the device input data over the network, allowing, for example, distributed applications and scenarios.

As opposed to the broad hierarchy of VRPN, the abstraction layer of *DEVAL* [22] establishes a deep hierarchy that puts more emphasis on the exchangeability of devices. Both approaches are based on abstracting from concrete devices and introducing hardware or device classes. These approaches have limitations when it comes to multimodal exchangeability of interaction techniques. In contrast, *DEMIS* [15] relies on events. It also accounts for multi-level composite events and is placed between the operating system and an application.

Frameworks such as emphMidas [24] focus on multi-touch and further enhance the I/O abstraction. In terms of distributed output and cross-device interaction, *PolyChrome* [1] can be used to seamlessly connect multiple devices for collaborative, web-based visualizations. Systems that want to support multi-device interaction can benefit from the Device Indepentent Architecture [5]. Similar to our system, the authors propose to decouple devices from applications in order to adapt to the given environment. The work around the *Virtual Interactive*

Namespace (VINS) [26] provides a distributed memory space that permits the reuse and exchange of various interactive techniques, which also enhances the development of reusable interaction components. Another library that supports designers and researchers with regard to the development of novel interaction techniques is *Squidy* [18]. It unifies various device drivers, frameworks, and tracking toolkits and exposes a visual design environment to increase the overall ease of use.

In order to establish our interaction event types, we have chosen the contributions of Card et al. [4] and Mackinlay et al. [19] as our starting point. Their work in this area focuses on the design space for input devices. One key idea is to split a device into a set of atomic capabilities, e.g., a mouse wheel and mouse buttons and the movement sensor in the case of a mouse. These capabilities are captured by a taxonomy consisting of classes such as 1D-3D motion or rotation. Hence, mouse movement would be classified as 2D motion on the x- and y-axes. In contrast to that rather mechanical point of view, the work of Javob et al. [14] focuses more on the perceptual structures of interaction tasks.

There is also work on other taxonomies dealing with less traditional I/O techniques. For example, one might consider gesture recognition. Here, *Proton* [17] proposes a regular expression-based classification of touch input. The work of Nebelig et al. [21] evaluates user-defined Kinect gestures and speech commands for the interaction with a wall-projected web browser. For mobile devices, user-defined gestures are composed into a motion gesture taxonomy of Ruiz et al. [23]. Widgets are another important approach to generating user input. One example of widget classification with a focus on 3D tasks can be found in the work of Dachselt and Hübner [6].

We do not merely aim to classify devices but also to establish exchangeable connections to applications. For that reason, one has to deal with the application side of the interaction pipeline as well. Applications can have a variety of interaction tasks to be performed. The following six tasks, mainly suited for 2D, were proposed by Foley [9]: select, position, orient, path, quantify, and text. For 3D interaction, five basic interaction tasks were introduced and refined by Bowman [2, 3]: navigation, selection, manipulation, system control, and symbolic input.

2.2 UI Adaptation

Our scenario involves varying environments, tasks, and user preferences. The task of adapting a user interface to such constraints can be tackled in multiple ways. For instance, users can assign the output of directly connected devices to application functions with the visual editor *ICON* [7]. To a limited degree, input transformation is possible as well, but requires a skilled user to perform the configuration. *SUPPLE* [10] formalizes the UI configuration problem and focuses on the graphical aspect of automated UI generation. Its successor, *SUPPLE++* [11], adds support for physically disabled users by including user models. Kim et. al [16] introduce interaction layering and abstraction based on device capabilities to overcome the issue with different interaction environments. UI adaptation also plays an important role in the automotive industry, driven especially by the amount of external infotainment possibilities as discussed in [20].

3 MORPHABLEUI

We start off with outlining an abstract model for UIs and user interaction in general. We enhance the construct by enabling dynamic transformation of interaction events via the split, merge, and cast operators. Based on that model, we describe our novel approach of generating admissible UIs using a hypergraph-based algorithm. We conclude by presenting an implementation of these concepts and offering a user- and developer-friendly way to establish dynamic connections between arbitrary applications and interaction devices.

3.1 Model

3.1.1 Events, Capabilities, and Requirements

Different interaction devices can be used to perform the same user task. In our medical example, the visualization application allows users to move, i.e., pan, the dataset, which can be achieved by moving the mouse and also by swiping over a smartphone touchscreen. From a more abstract point of view, what the mouse and the smartphone provide is the ability to generate *interaction events* of a specific type that are sent to and interpreted by the application. Both devices generate the same *type* of interaction event, precisely, a two-dimensional motion event. Because the mouse and the smartphone provide the means of generating interaction events of the same type, they can be exchanged with respect to the task to be performed.

The device characteristics or *capabilities* describe the type of generated or processed interaction events. Input capabilities generate interaction events triggered by the user, whereas output capabilities process interaction events received from the application such as video output. Note that some devices, e.g., smartphones, have input as well as output capabilities.

The capability classification includes low-level types of interaction events, e.g., `Firing Event` or `2D Position`, as well as higher-level types such as `3D Manipulation`. An example classification illustrating both input and output capabilities of a smartphone is

Capability	Interaction event type
Pinch gesture	`Zoom Event`
Gyroscope	`3D Rotation`
Slider widget	`1D Motion`
Touchscreen position	`2D Position`
Touchscreen display	`Video`
Voice recognition	`Text`

Table 1: Excerpt of the capabilities of a smartphone.

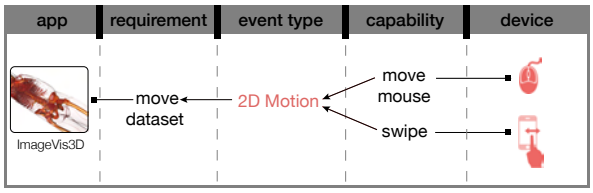


Figure 2: By introducing requirements and capabilities, the basic model decouples applications from devices. Both sides are associated with the corresponding interaction event types. In this example, moving the mouse can be used to pan the dataset. Since the swipe gesture is associated with the same interaction event type, these two interaction techniques can be exchanged.

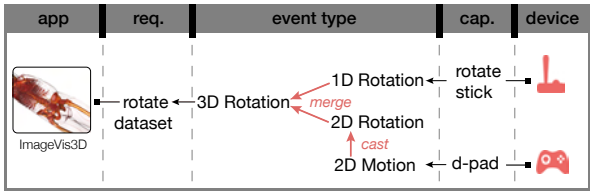


Figure 3: Adding the novel split, merge, and cast operators allows the transformation of generated interaction events and combination of different devices to perform a task. Hence, rotating the dataset can be achieved by a combination of the directional pad (d-pad) of a gamepad and the stick rotation of a joystick.

given in Table 1. A `Zoom Event` can also be regarded as an event of type `1D Motion`. However, the pinch gesture capability of a smartphone is tailored to accomplish the very specific task of zooming in or out, which is why we associate it with that higher-level event type. As explained in the next section, such a fuzzy specification is not an issue since event types can be transformed into other types if certain criteria are met.

Analogous to capabilities of devices, applications have *requirements* for specific user tasks. Each one is tied to an event type. Viewing devices and applications in this way allows the exchange of devices if their capabilities cover the requirements of the application as shown in Figure 2. We call an admissible connection between a capability and a requirement a *wiring*. Since an application usually consists of multiple requirements, the complete user interface can be formally defined as a set of wirings.

Another aspect to be mentioned regarding the user experience is the I/O data sensitivity and range. These additional properties can be provided on both the application and device sides to enable automated unification inside the framework that internally uses a unit hypercube, which often results in improved interaction compared to raw input that might differ significantly across devices.

3.1.2 Dynamic Event Modification

In addition to panning, we now want to rotate our dataset, which requires a `3D Rotation` event. One might use a gyroscope in a smartphone to generate the necessary input. However, one also could combine, i.e., merge, different lower-dimensional input capabilities. Hence, the definition of a wiring must be extended to also include connections between one requirement and multiple capabilities. The latter have to generate interaction events that can be transformed to yield a single event matching the application requirement.

We suggest three types of operations on interaction events that allow such transformations: cast, split, and merge. The *cast operator* transforms the semantics of an interaction event if possible. In the case of a `Zoom Event`, one is able to cast it to `1D Motion`. The *split operator* splits one event into multiple, in most cases lower-dimensional, events. Hence, a 3D gyro sensor can be used for panning a picture in 2D by splitting the underlying `3D Motion` capability into `1D Motion` and `2D Motion`. The *merge operator* is its inverse and merges multiple interaction events into one. An example transformation pipeline for the `3D Rotation` requirement is depicted in Figure 3.

3.2 Graph

Being able to transform device I/O according to the three introduced operators clearly enhances the UI design space. This section tackles the issue of computing such wirings. First, a number of different representations for the interaction event types and their interconnection are discussed. Second, we present an iterative algorithm that proposes admissible wirings for a given requirement.

Taking interaction events as input, the operators execute a certain transfer function and return the corresponding interaction event (or events, in the case of a split operation) as a result. From the point of view of an interaction event, operators are perceived as incoming, if that event is the result, or outgoing, if that event is the input.

One way to project this model onto a data structure is to use trees with the event types as vertices and operators as edges. Another approach is to use context-free grammars with event types as symbols and operators as production trees. Intuitively, both approaches share the same computational logic: one starts at the type of the application requirement and examines all possible decompositions. At this point, two major drawbacks can already be observed. First, both representations contain duplicates of event types since each one can have multiple outgoing and incoming operators. As a result, the representation is difficult to maintain since one has to care about all production rules or trees if a type or operator is added or removed. Second, the need to account for all possible decompositions leads to an exponential runtime of the algorithm, which is a problem in cases with a mentionable number of devices and operators.

We design a hypergraph with event types as vertices and operators as hyperedges. Informally, this generalized graph form is needed because the split and merge operators represent a 1-to-N connection and involve more than two vertices. Hyperedges allow N-to-M connections and are a feasible data structure for our task. One additional

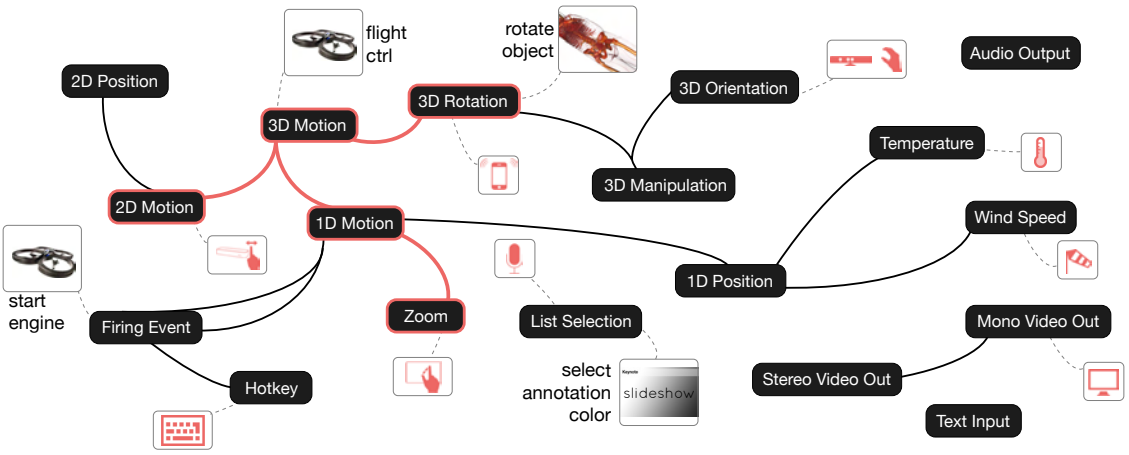


Figure 4: A subset of the established hypergraph. Event types are captured as vertices whereas hyperedges represent the operators. To maintain clarity, a number of edges and vertices are omitted. The proposed iterative algorithm uses device tokens that traverse the hypergraph until the requirement vertex is reached. The result is a subgraph representing the wiring between device capabilities and a requirement. One example of a wiring is highlighted.

concept based on the work in [12] is utilized, the so-called *backward* and *forward arcs*. Both are special types of directed hyperedges, either 1-to-N (forward) or N-to-1 (backward). Hence, a forward arc precisely expresses the layout of the split operator, and a backward arc represents a merge operation. Thus, the task of computing admissible UIs can be completed by computing a sub-graph connecting the vertex associated with the application requirement to one or more vertices representing device capabilities as depicted in Figure 4.

Note that the length of a path between two vertices directly corresponds to the resulting transfer function applied on the I/O data. Thus, a large distance, i.e., a large number of required operators, corresponds to a less direct mapping. The distance aligns with one's intuition since using three `1D Motion` events to accomplish a `3D Motion` task is less direct than using a single `3D Motion` event. Based on that property, an iterative algorithm that presents possible wirings ordered by ascending distance between requirement and device capabilities is beneficial. Hence, a user would first receive a number of adjacent solutions and demand further solutions if needed.

Our key idea is to use tokens commonly known from Petri Nets. Each token represents a device and is initially placed at the corresponding vertex. For example, a token for the swipe gesture will start in the `2D Motion` vertex. Tokens can be moved over edges to adjacent vertices if the traversal requirements outlined in Table 2 are met.

Possible traversals are executed sequentially, ordered by their cost. Similar to Dijkstra's shortest path algorithm, the cheapest traversal is estimated by computing the distance we already traveled as formalized in Table 2. The approach is summarized in Algorithm 1. Tokens arriving at the vertex corresponding to the requirement carry a valid wiring since the token history stores the sequence of executed traversals. In this way, solutions are presented to the user step by step. Again, later proposals indicate a less direct transfer function is needed to transform the I/O data required by the application. To sum up, the main advantages of the presented approach include the iterative solution generation, the in-place search with a data structure without duplicated event types, and the amortized polynomial time and space of the algorithm.

Finally, we establish a way to validate external, e.g., handcrafted, assignments of device capabilities for a requirement. For this purpose, the same algorithm can be employed. The corresponding device capability tokens are inserted, and the algorithm executed until a solution is found, no further traversals can be executed, or a step limit is reached. If the algorithm finds a solution, the demanded mapping is admissible, and the wiring including the required operator chain is returned. Note that this procedure allows for black box proposals consisting of the endpoints—requirement and capabilities—without the need to provide the complete operator sequence.

3.3 Implementation of MorphableUI

We addressed the distributed multi-device design issue by developing an interaction model and a corresponding algorithm that computes admissible wirings for given application requirements. In the following, we demonstrate our implementation of MorphableUI to prove the established concepts. We introduce three main components: *Gates* that serve as entry points for application and device developers. A *server* that maintains the interaction topology and provides external services, and the *MasterUI*, a mobile frontend that builds on such a service and allows users to select and configure UIs.

3.3.1 Gates

A MorphableUI gate is a C-library that allows users to plug applications and devices into the interaction topology spanned by our framework. The gate component

Algorithm 1 Iterative computation of admissible wirings. The algorithm returns tokens that arrive at the requirement vertex. The corresponding sequence of operators can be extracted from $hist(t)$. Traversal rules and definitions can be found in Table 2. The algorithm sequentially executes the next cheapest traversal. After an execution and the resulting token movement, traversals of the affected vertices have to be updated.

Input:
application requirement r
device capabilities $c_1, ..., c_n$

Initialization:
for all c_i **do**
  insert new t into corresponding v
end for
create empty $TraversalList$
mark requirement vertex as v_r
for all e **do**
  compute cheapest $trav_{V \to W}$ on e (see R)
  add $trav_{V \to W}$ to $TraversalList$
end for

Iteration:
repeat
  exec. cheapest $trav_{V \to W}$ in $TraversalList$ (see R)
  for all $v \in V, W$ **do**
    for all e, e incident to v **do**
      remove $trav_{V \to W}$ associated with e from $TraversalList$
      compute cheapest $trav'_{V \to W}$ on e
      add $trav'_{V \to W}$ to $TraversalList$
    end for
  end for
until new t arrives in v_r
return t

comes with a simple API that allows users to send and receive interaction events as shown in the Listing 1. An application that demands `3D Rotation` only has to call such a receive function or register a callback to obtain incoming events. On the other side of the pipeline, e.g., the gyro sensor of a smartphone constantly pushes its captured rotation events via a send function.

Gates gather the information about application requirements and device capabilities from a developer-provided json file. Our implementation in plain C allows the use of the same gate implementation on desktops, mobile (iOS, Android), and other platforms such as Raspberry Pi. To faciliate the integration into modern software, a set of wrappers in other languages is available. The wrappers expose the same API and are available in languages such as Python, JavaScript, Java, C++, Objective-C, and Go. In terms of interaction event streaming performance, we point out that the gate-to-gate streaming is executed directly, i.e., without routing data over the server component presented in the next section.

Traversal rules R

Definitions and notation
- v, w : vertices, e : edge, V, W : sets of vertices
- $trav_{V \to W}$ traversal on edge connecting V and W
- traversal types: $split_{v \to W}$, $cast_{v \to w}$, $merge_{V \to w}$
- t a token with history $hist(t)$ of executed traversals
- t associated with one or more (after merging) device capabilities c
- $cost(t) = |hist(t)|$

Candidate tokens for a traversal $trav_{V \to W}$
- all t in $v \in V$ not yet visited any $w \in W$
- merge constraint: one t from each $v \in V$ required and selected tokens must not be associated with the same c (prevents merging a capability with itself)
- cheapest $trav_{V \to W}$ (not necessarily unique) defined as: $min(\sum cost(t) \mid t$ participating in $trav_{V \to W})$

Executing a traversal $trav_{V \to W}$
- $split_{v \to W}$: $\forall w \in W$: insert duplicate t_d of t_{src} in w
- $cast_{v \to w}$: insert duplicate t_d of t_{src} in w
- $merge_{V \to w}$: insert new t_n in w, $\forall t_{src}$: add $hist(t_{src})$ to $hist(t_n)$
- $\forall t (t_d$ or $t_n)$ add $trav_{V \to W}$ to $hist(t)$
- $\forall t_{src}$: if visited all adjacent v and $\nexists$ outgoing merge edge: $delete(t_{src})$
- *note:* the second condition is needed since a potential merge candidate might arrive later

Table 2: Our Algorithm 1 operates on tokens. They initially represent device capabilities and are moved in a hypergraph on edges standing for operators between vertices representing the interaction event types.

3.3.2 Server

While gates are spread over the network, their operability depends on a central server that is responsible for the environment coordination. Precisely, the server contains a memory-efficient C++ implementation of Algorithm 1, maintains user sessions, and keeps track of available gates. The server behaves as a broker between the gates and the users. It exposes necessary information about available applications and devices gathered from the gates to the users and configures gate streaming pipelines according to the user-definded interfaces. Apart from this UI configuration functionality, further explained in the next section, the server exposes a set of external services available for developers. For instance,

```
// initialization
Gate gate("ImageVis3D.json");
gate.start();
// runtime
Event evt = gate.receiveEvent("Pan␣dataset");
// something inside the target software
translate(glm::vec3(evt.x, evt.y, 0), data);
```

Listing 1: Example integration in C++. The gate is initialized with a json file containing a list of requirements or capabilities. During runtime, the target software polls or sends interaction events.

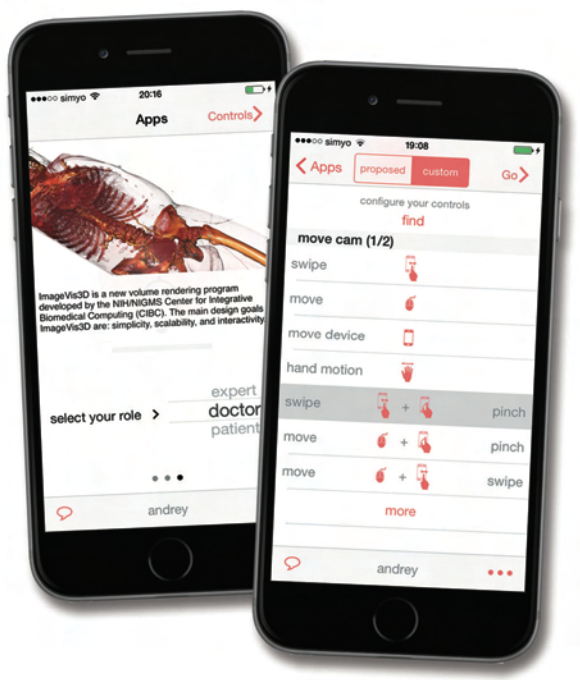


Figure 5: The MasterUI is a mobile frontend for our system that allows to select and customize UIs. The application selection screen already prototypes the role feature addressed in future work. The right image displays possible assignments for a given requirement.

REST interfaces are provided for both UI generation and interaction environment monitoring.

Assume that the server received a UI configured by a user via our frontend. According to our model, the UI consists of one wiring for each application requirement, while each wiring represents an I/O processing pipeline with a sequence of operators. This information is sent to all participating gates that then dynamically set up the necessary gate-to-gate streaming connections. The operator chain is always placed on the receiving side since data might be incomplete prior to that point. Hence, in the case of an input requirement, the operator chain resides in the application gate and vice versa.

3.3.3 UI Configuration App

One of the services provided by the server is to allow configuration and launch of user-defined interfaces. To deliver a user- and developer-friendly contribution, MorphableUI comes with a default mobile frontend, the *MasterUI*, depicted in Figure 5. This app guides users through the UI generation pipeline and allows on-the-fly customization during runtime, which is beneficial for, e.g., rapid prototyping tasks. This component is designed as a personal assistant that behaves according to the bring-your-own-device paradigm. Hence, everyone is able to use their private smartphone to create and apply desired UIs.

An example configuration process is depicted in Figure 5. First, the user has to choose the application he or she wants to control. In a second step, the UI is assembled. The straightforward way is to configure each application requirement manually. Hereby, wirings are requested from the hypergraph algorithm in an iterative way until the user sees a satisfying device assignment.

Remember that the proposal ordering corresponds to the length of the involved operator chain and thus reflects the number of needed event transformations between the capabilites and the application requirement. Instead of manually configuring each wiring, users are also able to request automated proposals for complete UIs and choose between or reconfigure them.

The ability to obtain automated UI proposals is based on stored information about previous usage, i.e., on what the user already designed for this or similar applications. Similarity, then, is defined by the percentage of equal requirement types. Intuitively, there is a chance that requirements associated with the same type of interaction event behave analogously. Thus, we implicitly port UIs across applications by generating such proposals. This approach also accounts for interfaces designed by others since it turned out to be a good starting point compared to blank initialization. One of our future goals is to enhance this automated proposal and learning ability to anticipate users' needs and minimize the UI configuration efforts.

4 INTEGRATING MORPHABLEUI

To demonstrate how the theoretical model and its realization behave in the real world, we have added support for a set of devices and applications and evaluated the integration efforts of our framework in external projects. A few lines of code suffice to enable full access to the MorphableUI interaction features. The Listing 1 provides an overview of the necessary steps including setup and runtime. Note that the appications does not need to know the available devices at all nor to restart or recompile if a new device becomes available.

4.1 Sample Devices and Applications

4.1.1 Device Support

Our prototype covers conventional desktop environments, joysticks, gamepads, Kinects, Leap Motions, monitors and mobile displays for mono video output, and head-mounted displays for stereo video output. The underlying video streaming relies on a JPEG-encoded frame transmission, i.e., each frame is packed into an interaction event and transported to the output device. Furthermore, MorphableUI supports iOS and Android smartphones and tablets. These devices expose capabilities such as swipe and pinch gestures, gyroscope and accelerometer sensors, speech input, and widgets such as sliders and virtual joysticks. To demonstrate the range of possible use-cases, smart home sensors for temperature, wind speed, and air pressure were also integrated.

4.1.2 Application Example: ImageVis3D

The volume rendering software *ImageVis3D* [8] scales to very large biomedical datasets. It already accounts

Requirement	Interaction event type
Rotate dataset	`3D Rotation`
Pan dataset	`2D Motion`
Resize dataset	`Zoom`
Toggle between 1D-TF and Iso	`Toggle`
Rotate clipping plane	`3D Rotation`
Set smoothstep function for TF	`2D Position`

Table 3: A set of ImageVis3D requirements. The one-dimensional transfer function is denoted by 1D-TF and isosurface rendering by Iso.

for heterogeneous environments by being able to run on everything from mobile devices to high-end graphics workstations. To allow such flexibility on the I/O side, we have connected the software to our framework by capturing a set of basic functionalities as shown in Table 3.

From the captured interaction tasks, manipulating the transfer function was of increased interest for the developers of ImageVis3D. One surprisingly intuitive interface was moving the hand over the leap motion and changing the inflection point of the smoothstep function by lifting or lowering the hand. Alternatively, rotating one's hand was also rated as intuitive for changing the slope of the smoothstep function.

4.2 Developer Survey

To gain feedback on our approach, we asked nine application developers (all male) interested in MorphableUI to fill out a questionnaire after the first successful integration of our system into their target software, including both academic and industrial collaborations to cover a wide sample range. The questions included both subjective topics (difficulty of integrating the software, required support) and objective topics (needed development time for integrating the libraries into their software including glue code, time for defining the requirements/capabilities). The subjective questions were answered via a 7-point Likert scale, with 1 meaning very easy/none and 7 indicating very hard/always. The objective questions used minutes as a scale, since they targeted development efforts. Complementary questions about the programming experience and age of the participants were designed to provide hints about possible side effects for inexperienced users.

Our results show that the integration of MorphableUI could be done in less than one hour in all cases, but most participants required less than 30 minutes. The design time for capabilities/requirements fluctuated more, linearly depending on the amount and complexity of the targeted interaction. For the question regarding the difficulty of integrating MorphableUI into existing software, we see a mean value of 2.0 with a standard deviation (SD) of $\sigma = 0.71$. Hence, the developers found the process easy and encountered no major difficulties. This result is further strengthened with the outcome for the question concerning the required support for integrating the software: it shows a mean value of 2.0 with a SD of $\sigma = 1.0$, meaning the developers only needed little support. We cannot conclude that the developer experience had a direct impact on the integration or requirement/-capabilities design time. Hence, the complexity of the target software seems to be a more prominent factor.

	Difficulty of Integration	Needed Help	Time for Integration	Time for Requirements	Exp.
P1	2	2	20 min	30 min	3 yr.
P2	3	3	40 min	30 min	1 yr.
P3	1	1	5 min	5 min	7 yr.
P4	2	1	15 min	10 min	2 yr.
P5	3	2	30 min	5 min	4 yr.
P6	2	4	40 min	20 min	6 yr.
P7	2	2	20 min	5 min	4 yr.
P8	2	2	30 min	5 min	4 yr.
P9	1	1	10 min	10 min	3 yr.
mean	2.0	2.0	$23.\overline{3}$ min	$13.\overline{3}$ min	$3.\overline{7}$ yr.
sd	1	1	12.5 min	10.61 min	1.86 yr.

Table 4: P1 to P7 were integrations of MorphableUI into existing applications, P8 and P9 added new interaction devices to our system. In detail, P1-P4 were interactive 3D visualization tools, P5 an interactive physical simulation, P6 a connection to the FMI standard, P7 the integration into OgreVR, P8 connected the Community Core Vision, and P9 added support for the Leap Motion.

5 BENEFITS AND LIMITATIONS

A common question is whether MorphableUI is beneficial for a particular application. Despite that results from preliminary user studies indicate high acceptance, this topic requires further discussion. On the one hand, mapping a complex software such as Photoshop with hundreds of different tasks does not seem feasible with the proposed technique. First, the UI generation process will generally consume more time, and reassigning certain controls will often result in scrolling through large lists compared to, for example, browsing well-structured settings menus. Second, applications that are tightly coupled to a specific environment or device setup often cannot take advantage of the offered I/O exchangeability. On the other hand, applications often have a set of basic functionalities that are accessible in different environments and can be controlled in multiple ways depending on the use-case. In the Photoshop example, users still might want to control panning and zooming via a tablet with the non-dominant hand.

Hence, we recommend combining MorphableUI with traditional hard-wired interfaces. That is, we suggest using our system to cover only a small subset of requirements where device exchangeability is expected to be important. In the case of our ImageVis3D scenario, we recommend users stick to a traditional UI for tasks such as opening a file and rely on MorphableUI for object manipulation or the streaming of the video output. During development, prototyping tasks benefit massively from the effortless integration, as the system allows to try out a plethora of I/O devices out of the box.

6 SUMMARY AND FUTURE WORK

The paper established a requirement- and capability-based model for distributed, multimodal interaction. We

introduced a classification building upon interaction event types and expressed their relations by three operators: split, merge, and cast. This formalization allows higher-order I/O data transformation and enhances the exchangeability compared to other approaches. The problem of computing admissible UIs by generating wirings for each requirement is tackled by a token-based, iterative algorithm working on a hypergraph. The vertices correspond to the interaction event types, and edges represent the operators. The generated wiring proposals are ordered by the length of the resulting operator chain. Hence, the order expresses the amount of performed I/O data transformation.

The implementation consists of three main components. The gate is a library that serves as an entry point for applications and devices. During runtime, I/O data is streamed directly between the gates over the network. The server monitors the devices and applications and exposes services such as the UI configuration. MasterUI is a mobile frontend built upon such a service. It allows users to dynamically configure their interfaces and receive notifications about changes in the interaction topology. Finally, support was added for a set of sample devices and applications to showcase the effortless integration of our system which particularly enhances the area of rapid prototyping of multimodal interfaces.

There are a number of different issues to be targeted in the future. One goal is to increase the number of supported devices such as the Microsoft HoloLens, which is mainly an engineering task. At this point, it might be of interest to extend the existing model by hardware characteristics of the input devices. Considering widgets as I/O capabilities, a sophisticated arrangement would be beneficial. For now, the system does not have any hierarchical concepts and places the widgets, such as virtual joysticks, at predefined positions. To fully support that kind of interaction, the UI configuration pipeline has to be extended to deal with layout settings.

Another idea is to arrange requirements into roles on the application side as already prototyped in Figure 5. In our brain stimulation example, one would differ between the doctor and patient. The latter role has limited interaction requirements allowing, for example, only to panning and rotating the dataset. Another user-related feature that will be included in future work is security and authentication. One use-case is to prevent unauthorized I/O device access to a set of private devices limited to one particular user.

In this paper, we mainly focused on I/O exchangeability and enabled a novel approach for multimodal, distributed interaction and rapid prototyping. One of our next goals is to measure and enhance user experience in MorphableUI by conducting usability studies. Clearly, our framework can also be used to provide interfaces that are not very user-friendly. For this reason, we plan to combine the UI generation with a sophisticated online learning algorithm to further improve the UIs being proposed automatically based on prior knowledge. In addition, multi-user setups will be focused more since the framework does not impose any limitations on the number of users for one application.

The set of interaction events that we used for the capability and requirement classification does not pretend to be complete. Further refinement is needed depending on the application area and the use-case. To tackle this issue, we are developing a MorphableUI tool chain. The chain will include a GUI-based hypergraph modification tool that allows the addition of new types of events without the need to touch or (re-)compile code. Also, a web application that facilitates monitoring the interaction environment and assists developers and administrators would further enhance the framework. The main issue, therefore, is to find a compact and meaningful graphical representation of the environment including device locations, active user interfaces and the corresponding wirings between devices and applications.

REFERENCES

[1] S. K. Badam and N. Elmqvist. “PolyChrome: A Cross-Device Framework for Collaborative Web Visualization”. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’14. Dresden, Germany: ACM, 2014, pp. 109–118. ISBN: 978-1-4503-2587-5.

[2] D. A. Bowman et al. *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004. ISBN: 0201758679.

[3] D. A. Bowman et al. *Interaction Techniques For Common Tasks In Immersive Virtual Environments - Design, Evaluation, And Application*. 1999.

[4] S. K. Card, J. D. Mackinlay, and G. G. Robertson. “The design space of input devices”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’90. Seattle, Washington, United States: ACM, 1990, pp. 117–124. ISBN: 0-201-50932-6.

[5] J. Chmielewski and K. Walczak. “Application Architectures for Smart Multi-device Applications”. In: *Proceedings of the Workshop on Multi-device App Middleware*. Multi-Device ’12. Montreal, Quebec, Canada: ACM, 2012, 5:1–5:5. ISBN: 978-1-4503-1617-0.

[6] R. Dachselt and A. Hübner. “A Survey and Taxonomy of 3D Menu Techniques”. In: *Proceedings of the 12th Eurographics Conference on Virtual Environments*. EGVE’06. Lisbon, Portugal: Eurographics Association, 2006, pp. 89–99. ISBN: 3-905673-33-9.

[7] P. Dragicevic and J.-D. Fekete. “Input Device Selection and Interaction Configuration with ICON”. In: *Proceedings of the HCI01 Conference on People and Computers XV*. Springer, 2001, pp. 543–558.

[8] T. Fogal and J. Krüger. "Tuvok, an Architecture for Large Scale Volume Rendering". In: *Proceedings of the 15th International Workshop on Vision, Modeling, and Visualization*. 2010.

[9] J. D. Foley, V. L. Wallace, and P. Chan. "The human factors of computer graphics interaction techniques". In: *IEEE Computer Graphics and Applications* 4.11 (1984), pp. 13–48. ISSN: 0272-1716.

[10] K. Gajos and D. S. Weld. "SUPPLE: Automatically Generating User Interfaces". In: *Proceedings of the 9th International Conference on Intelligent User Interfaces*. IUI '04. Funchal, Madeira, Portugal: ACM, 2004, pp. 93–100. ISBN: 1-58113-815-6.

[11] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld. "Automatically Generating User Interfaces Adapted to Users' Motor and Vision Capabilities". In: *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST '07. Newport, Rhode Island, USA: ACM, 2007, pp. 231–240. ISBN: 978-1-59593-679-0.

[12] G. Gallo et al. "Directed hypergraphs and applications". In: *Discrete Appl. Math.* 42.2-3 (Apr. 1993), pp. 177–201. ISSN: 0166-218X.

[13] M. Hachet, P. Guitton, and P. Reuter. "The CAT for Efficient 2D and 3D Interaction As an Alternative to Mouse Adaptations". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '03. Osaka, Japan: ACM, 2003, pp. 225–112. ISBN: 1-58113-569-6.

[14] R. J. K. Jacob et al. "Integrality and Separability of Input Devices". In: *ACM Trans. Comput.-Hum. Interact.* 1.1 (Mar. 1994), pp. 3–26. ISSN: 1073-0516.

[15] H. Jiang, G. D. Kessler, and J. Nonnemaker. "DEMIS: A Dynamic Event Model for Interactive Systems". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '02. Hong Kong, China: ACM, 2002, pp. 97–104. ISBN: 1-58113-530-0.

[16] S. J. Kim et al. "Adaptive interactions in shared virtual environments for heterogeneous devices". In: *Computer Animation and Virtual Worlds* 21.5 (2010), pp. 531–543. ISSN: 1546-427X.

[17] K. Kin et al. "Proton: Multitouch Gestures As Regular Expressions". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, 2012, pp. 2885–2894. ISBN: 978-1-4503-1015-4.

[18] W. A. König, R. Rädle, and H. Reiterer. "Interactive Design of Multimodal User Interfaces - Reducing technical and visual complexity". In: *Journal on Multimodal User Interfaces* 3.3 (2010), pp. 197–213.

[19] J. Mackinlay, S. K. Card, and G. G. Robertson. "A semantic analysis of the design space of input devices". In: *Hum.-Comput. Interact.* 5.2 (June 1990), pp. 145–190. ISSN: 0737-0024.

[20] G. de Melo et al. "Towards a Flexible UI Model for Automotive Human-machine Interaction". In: *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '09. Essen, Germany: ACM, 2009, pp. 47–50. ISBN: 978-1-60558-571-0.

[21] M. Nebeling et al. "Web on the Wall Reloaded: Implementation, Replication and Refinement of User-Defined Interaction Sets". In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. Dresden, Germany: ACM, 2014, pp. 15–24. ISBN: 978-1-4503-2587-5.

[22] J. Ohlenburg, W. Broll, and I. Lindt. "DEVAL: a device abstraction layer for VR/AR". In: *Proceedings of the 4th international conference on Universal access in human computer interaction: coping with diversity*. UAHCI'07. Beijing, China: Springer-Verlag, 2007, pp. 497–506. ISBN: 978-3-540-73278-5.

[23] J. Ruiz, Y. Li, and E. Lank. "User-defined Motion Gestures for Mobile Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 197–206. ISBN: 978-1-4503-0228-9.

[24] C. Scholliers et al. "Midas: A Declarative Multitouch Interaction Framework". In: *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '11. Funchal, Portugal: ACM, 2011, pp. 49–56. ISBN: 978-1-4503-0478-8.

[25] R. M. Taylor II et al. "VRPN: a device-independent, network-transparent VR peripheral system". In: *Proceedings of the ACM symposium on Virtual reality software and technology*. VRST '01. Baniff, Alberta, Canada: ACM, 2001, pp. 55–61. ISBN: 1-58113-427-4.

[26] D. Valkov, A. Giesler, and K. H. Hinrichs. "VINS - Shared Memory Space for Definition of Interactive Techniques". In: *ACM Symposium on Virtual Reality Software and Technology (VRST 2012)*. ACM, 2012, pp. 145–153.

[27] S. Zhai and P. Milgram. "Quantifying Coordination in Multiple DOF Movement and Its Application to Evaluating 6 DOF Input Devices". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., 1998, pp. 320–327. ISBN: 0-201-30987-4.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Facial expression recognition using salient facial patches

Hazar Mliki
MIRACL-ENET'COM
University of Sfax
Tunisia (3018), Sfax
mliki.hazar@gmail.com

Mohamed Hammami
MIRACL-FSS
University of Sfax
Tunisia (3018), Sfax
mohamed.hammami@fss.rnu.tn

ABSTRACT

This paper proposes a novel facial expression recognition method composed of two main steps: offline step and online step. The offline step selects the most salient facial patches using mutual information technique. The online step relies on the already selected patches to identify the facial expression using an SVM classifier. In both steps, the LBP operator was used to extract facial expressions features. Through an extensive experiments on the JAFFE and KANADE databases, we have shown that our method, thanks to the salient selected patches, has the advantage of being much faster with a significant gain in recognition performance.

Keywords

machine vision, facial expression recognition, Mutual Information, LBP

1 INTRODUCTION

In recent years, there has been an increasing interest on facial expressions recognition as it is one of the most important cues to our emotional state [VTG+15]. In fact, by analysing the emotional state of one person, we can easily extract information about its mood, feeling and personality. Therefore, facial expressions recognition has been involved in many computer vision applications, like surveillance systems, human-machine interaction, gaming and remote monitoring of patients [SGA09]. Although the continued research interest on facial expressions topic, recognizing facial expression with a high accuracy remains a challenging task due to the variation of facial expressions across human culture and to the context-dependent variation even for the same person.

Developing an efficient facial representation from face images is a key step to succeed facial expression recognition task. Actually, facial expression recognition includes two main stages: the facial feature extraction and the classification strategy. Facial feature extraction consists of deriving features which maximize between class variations whereas minimize within class variation of facial expressions. Hence, facial expressions recognition performance depends heavily on the choice of features used by the classifier. Relying on the way how facial features are extracted for classification, previous methods for facial expression recognition can be classified into two main approaches: geometric approach and global approach.

Geometric approach is based on the shape and locations of facial components such as the mouth, the nose, the eyes and the eyebrows. Then, the different distances between feature points and the relative sizes of the major face components are computed to form a feature vector. For instance, in [LBA99b] [GD03], the authors applied a geometric position of 34 manually selected points and a set of Gabor wavelet coefficients at these points. Some other authors [Ham06] compute relative distances to encode the geometric distance variations. Other [SJD08] used the geometric feature extracted by Active Appearance Model to perform facial expression recognition. Geometry approach is more robust to scale, size, head orientation variation. However, it requires reliable facial feature detection, which is a challenging task. Thus, most of the above cited methods, mainly [LBA99b] [GD03], require a manual selection of facial points which is not suitable for the autonomy aspect of the method. Moreover, facial features are unable to encode facial texture change such as wrinkles and furrows which are important for facial expression modeling.
In contrast, global approach encodes the appearance texture of the whole face which includes wrinkles, bulges and furrows. In this context, image filters are applied to the whole face so as to extract facial appearance variation which usually generates a high-dimensional feature vector. Accordingly, some

subspace learning methods such as principal component analysis (PCA) [DC99] and its independent form (PCI) [DGG06] are frequently performed to build new low subspace representation of the original face image. Then, matching is performed within the new subspace.

To sum up, we notice that geometric methods provide good perceptive justification for facial expression recognition. However, they depend on the accurate detection of facial features and require space costs for computation. Nonetheless, global methods inspect the appearance face variations which make them powerful to extract the discriminative information. Taking all this into account, we introduce in this paper a new method for facial expressions recognition which belongs to the global approach.

The remaining parts of the paper are organized as follows: in section 2, we introduce the proposed method, then we discuss the experiments in section 3 and conclude this paper in section 4.

2 PROPOSED WORK

The proposed method is based on psychological studies [Mag07] which show that some facial muscles are responsible of facial expressions appearance. These facial muscles are mainly located around some facial features such as the mouth, the nose and the eyes. The proposed method aims to define automatically the salient facial patches responsible of the local facial appearance variations. The proposed method is composed of two steps: offline step and online step. Both steps are performed after locating the face region using Viola and Jones Face detector [VJ04]. The offline step selects the most salient facial patches using mutual information technique. The online step relies on the already selected patches to identify the facial expression using an SVM classifier. In both steps, the extraction of the feature vector is carried out by LBP operator. The choice of such an operator is motivated by the fact that the face can be perceived as a combination of micro-models (patches). Such process allows managing the local variations of the face mainly due to illumination variation. Figure 1 describes the proposed method.

Our main contributions are:

- Automatic selection of the most salient face patches including the most discriminant descriptors to recognize facial expressions. Unlike the existing works which used manual and unpresice regions selection methods [FJJ09] [ST08] [LP12], we introduced a new algorithm based on Mutual Information technique to select automatically the descriptive patches. The identification of such patches reduces the complexity of the proposed method and thus accelerates the recognition process.
- Genericity of the selected facial patches. In fact, these patches are independent from the face images database and the used descriptor.

2.1 The off-line Step

This step seeks to select the active salient patches which are responsible of the facial expression deformation and appearance. Thus, we computed the facial feature vector using LBP operator. Then, we adapted the mutual information technique to select the most discriminant patches.

To extract facial expression features, we used the texture information by applying the LBP operator [OPH96]. We choose this operator thanks to its simplicity of computation which allows analysing images in real time as well as its invariance to rotation and illuminations variations. The LBP features are fast derived in a single scan through the raw image, whilst still including enough facial information in a compact representation.
After detecting the face region, we converted it to a grayscale image and applied an elliptical mask to get rid of hair, neck and all the noise that can appear jointly with the face. Thereafter, for a 64×64 pixels face region [LFCY06], we divided it into 64 patches each one is sized of 8×8 pixels. Finally, we coded each patch with an LBP histogram of 256 bins.
The choice of the number of patches is discussed in the experimental section. Figure 2 shows the process of feature vector extraction.

The selection of the optimal actives patches is the key point in our solution as it defines the quality and the performance of our method. The assumption here is that some patches may be insignificant, correlated or irrelevant and consequently, it would be interesting to remove them from the recognition process.
We have adapted the mutual information technique to select patches involving the most discriminant information for facial expressions recognition task. The mutual information (also called cross-entropy or gain-information) is a method of features selection widely used to measure the stochastic dependence of two discrete and random features [Soo00]. The mutual information between two variables x and y is defined based on their joint probabilistic distribution $p(x,y)$ and the respective marginal probabilities $p(x)$ and $p(y)$ as follow:

$$I(X,Y) = \int_{\Omega_Y}\int_{\Omega_X} p(x,y) log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy \qquad (1)$$

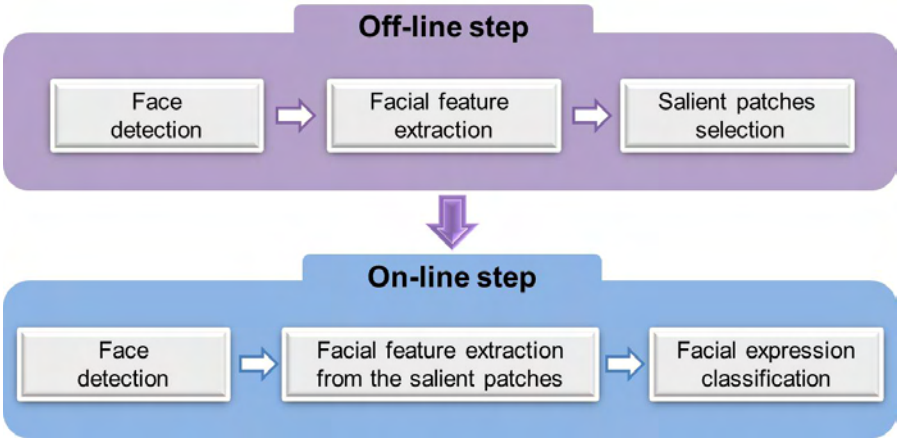


Figure 1: The proposed method for facial expression recognition

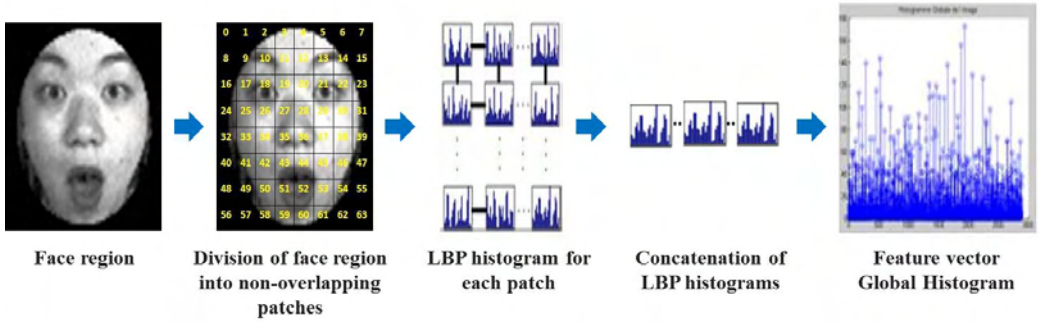


Figure 2: The process of feature vector extraction

Where Ω_X and Ω_Y are respectively the sample space of X and Y. Regarding $p(x)$, $p(y)$, and $p(x,y)$, they are respectively the probability density functions of X, Y, and (X,Y). In the pattern recognition applications, we expect a feature set that can remove the uncertainty of the class variable as much as possible. This can be achieved by finding a feature set $S_m = X_1, X_2, \ldots, X_m$ which jointly have the largest dependency on the target class c. This large dependency defines the Max-Dependency which has the following form in Eq.(2)

$$max \quad D(S_m, c) \tag{2}$$

Despite the theoretical value of Max-Dependency, it is often hard to get an accurate estimation for the multivariate density $p(x_1, \ldots, x_m)$ and $p(x_1, \ldots, x_m, c)$, because of the high-dimensional space. The high-dimensional space is due to the number of samples which is often insufficient and the multivariate density estimation which involves computing the inverse of the high-dimensional covariance matrix that is usually an ill-posed problem [PLD05]. So as the Max-Dependency criterion is hard to implement, an alternative is to select features based on maximal relevance criterion.

Actually, the max-Relevance creterion aims to select features that approximate with the mean value of all mutual information values between the individual features x_i and a class c. In fact, it searches features satisfying Eq.(3) which approximate $D(S_m, C)$ in Eq.(2) with the mean value of all mutual information values between the individual features x_i and the class c.

$$max \quad D(S_m, C), \quad D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i, c) \tag{3}$$

Our goal is to adapt the mutual information technique to select the most relevant patches. Thus, we calculated the relevance score of each facial feature using the criterion of maximum relevance. Based on the relevance score of each facial feature, we calculated the relevance score of each patch by summing up the relevance score of the patch features averaged by the number of features. This average score presents a measure of the patch pertinence. Finally, we sorted the relevance of patches relaying on their relevance score. An overview of Mutual Information adapted algorithm for regions selection is detailed below.

2.2 The on-line step

After the determination of the salient patches, this step is dedicated to the online-identification of facial expressions.

Algorithm 1 Relevent patches selection using Max-Relevance criterion

Variables:
N = Number of features
M = Number of patches
NF_{Patch} = Number of features per patch
FeatRel[] = Table of relevance per feature
PatchRel[] = Table of relevance per Patch
PatchRelSort[] = Sorted table of patches relevance
Sum = Sum of features relevance score per patch
1-Compute the relevance score for every feature.
for i = 1, . . . , N
FeatRel[i] = MaxRel($Feat_i$)
End_For
2-Compute the relevance score for every patch.
for k = 1, . . . , M
for j = 1, . . . , NF_{Patch}
Sum = Sum + FeatRel[j + (NF_{Patch} × k)]
End_For
PatchRel[k] = Sum / NF_{Patch}
End_For
3-Sort the RegRel table according to the patch relevance score.
PatchRelSort[] = sort(PatchRel);

Unlike the existing work [SG08] [SGM05] [FC15] which extract the feature vector from the whole image, we applied the LBP operator only the most discriminating patches to compute the LBP histogram. Thereafter, we concatenated the different LBP histograms to a single LBP histogram describing the overall appearance of the displayed expression as well as the spatial relationships between the selected patches. This LBP histogram involves information about the local distribution of the salient patches, such as the edges, the spots and the flat areas, to statistically describe the facial expression. Figure 3 describes the extraction of the feature vector from the relevant patches.

The generated LBP histogram provides a description of facial expression in three levels: the histogram labels involve information on a pixel-level, the summed labels of each patch describe the information on a region-level, and the concatenated histograms of each patch includes a description of the observed facial expression on a global-level.
In our work, we used the seven common classes of facial expressions: the neutral expression and the Ekman basic six expressions [Ekm72] : Neutral, Happiness, Fear, Disgust, Anger, Sadness and Surprise (cf. figure 4)

To build the facial expressions classifier, we processed with the SVM classifier [Vap98] as it allows a non-linear classification and it is independent from the size of the data space. Moreover, the robustness of the SVM classifier has already been proven in several studies of facial expressions recognition [BLFM03] [LBF+04]. As the SVM classifier takes binary decisions, a multi-class classification is performed by a cascading of binary classifiers with a scenario of vote. Thus, we described each face with a feature vector describing the preselected salient patches. Finally, the SVM classifier is applied to find out the optimal separation plan between facial expressions classes, and hence identify the corresponding facial expression class.

3 EXPERIMENTAL STUDY

Before presenting the results of the proposed method, we briefly describe the corpus and the used validation techniques.

3.1 Description of the corpus

The evaluation of the proposed method for facial expression recognition was performed on two databases:

- The JAFFE database (The Japanese Female Facial Expression) [LBA99a]: is widely used in the facial expressions research community. It is composed of 213 images of 10 Japanese women displaying seven facial expressions: the six basic expressions and the neutral one. Each subject has two to four examples for each facial expression.
- The KANADE database [KCT00]: is composed of 486 video sequences of people displaying 23 facial expressions within the six basic facial expressions. Each sequence begins by a neutral expression and finish with the maximum intensity of the expression. For fair comparison between KANADE and JAFFE databases, we selected from the KANADE database the first image (neutral expression) and the last three images (with the maximum intensity of the expression) of 10 people chosen randomly. Moreover, we selected the six basic facial expressions and the neutral one.

3.2 Techniques of validation

As a measure of validation, we used the Correct Classification Rate (CCR) of an expression defined as follow:

$$CCR = \frac{\text{Number of samples correctly classified as expression (E)}}{\text{Number of total samples with the expression (E)}} \quad (4)$$

The CCR is computed using the K-cross validation, with K = 10. Therefore, we segmented both of the image databases (JAFFE, KANADE) to 10 sets, and each time we use 9 sets for learning and keep the remaining set (not learned) for the test. We calculate the CCR for each test set and then we averaged these rates.

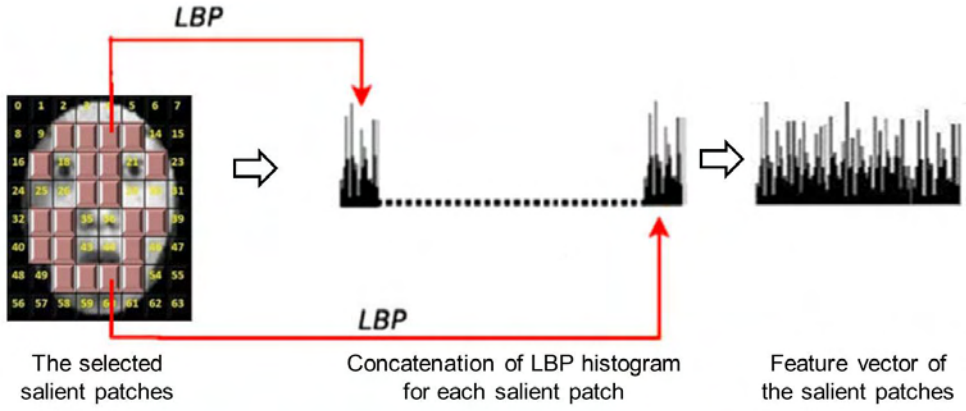


Figure 3: Feature vector extraction from the salient patches

Figure 4: The six basic expressions, from left to right : Anger, Disgust, Fear, Happiness, Sadness and Surprise.

3.3 Results of the proposed method

The experiments described in this section are justified by three reasons: (1) validate the choice of the number of division of the face image into patches , (2) validate the convenience of selecting the discriminant patches, and finally (3) compare the performance of our method with the most known works in literature.

3.3.1 First series of experiments

Through this experiment, we determine the number of the most appropriate division. Therefore, we tested different number of face divisions. Table 3.3.1 presents this experimental study.

The obtained results show that dividing the face image into 8*8 or 9*9 patches leads to the same CCR (93.89%). We opted for 8*8 divisions since it has the smallest dimension feature vector.

3.3.2 Second series of experiments

To select the most salient patches for facial expression recognition, we examined the evolution of the CCR through the number of the selected patches. Figure 5 shows this evolution.

Based on this assessment, we perceive how the CCR increases rapidly with the patches having the highest relevance score. In fact, we achieved the best CCR (93.89%) using only 21 patches. These patches are mainly located around the areas of the mouth, the eyes, the eyebrows and the nose (cf. figure 6) which validates the psychologists studies [Mag07].

To validate the relevance of the selected patches, we examined their independency from the database and the used descriptor. Therefore, we first applied our method of salient patches selection on a second images database: The KANADE database. The selected patches are shown in Figure 7 (b).

As shown in Figure 7, our method of patches selection produced 25 patches. Among the 25 selected patches, 21 are the same as those selected in the JAFFE database (cf. Figure 7 (a)). These results show an important overlap between the selected patches in JAFFE and KANADE databases. This proves the independency of the selected patches from the database and hence the genericity of our facial expression recognition method.

Besides, to study the independency of the selected patches from the used descriptor, we applied our method of salient patches selection on JAFFE database using the DWT (Discrete Wavelet Transform) descriptor. The choice of DWT operator rely on its several advantages mainly its simplicity of computation which allows analyzing real-time images as well as its invariance to illumination variations. Such an operator has been widely exploited in the context of facial expression recognition [ZZG04] [CW02] [MS00]. In fact, the DWT analyzes the image in different resolution levels using a low-pass and a high-pass filters. By applying

Number of patches	5×5	6×6	7×7	8×8	9×9
Feature vector size	12800	18432	25088	32768	41472
CCR	88.73%	90.61%	92.95%	93.89%	93.89%

Table 1: CCR based on the number of the patches

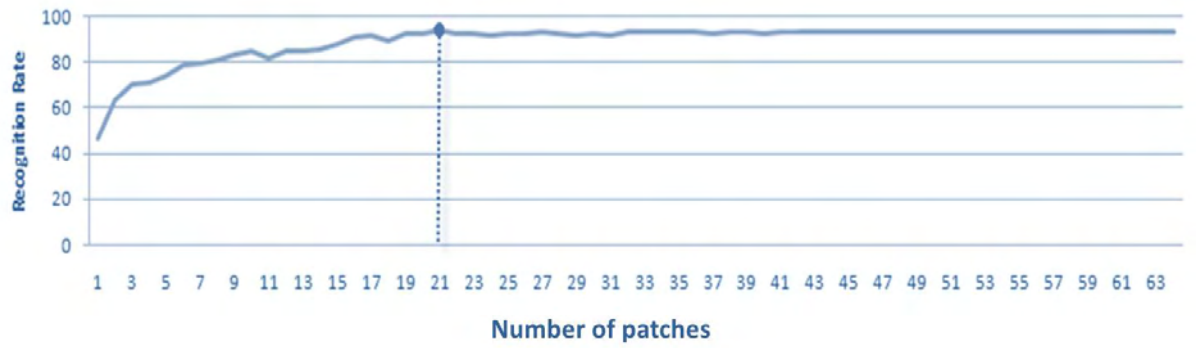


Figure 5: Evolution of the CCR through the number of the selected patches

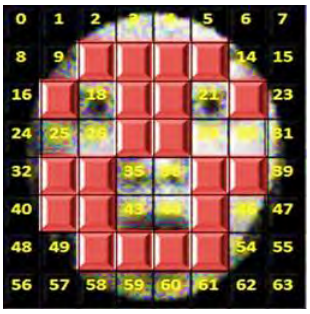


Figure 6: The selected patches

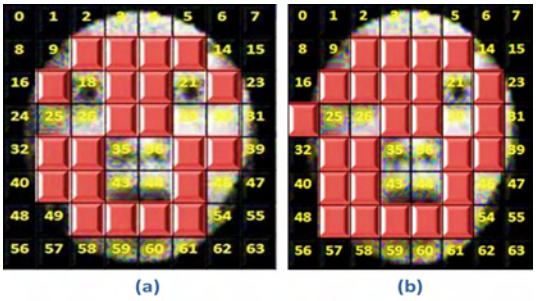


Figure 7: The selected patches from the JAFFE database (a) and the KANADE database (b)

the DWT operator on JAFFE database, 25 patches were selected (Figure 8 (b)).

According to Figure 8, we find out that among the 25 selected patches, 21 are the same as those selected with the LBP operator (cf. Figure 8 (a)). This overlap between the selected patches shows the independency of the selected patches from the used descriptor and thus the genericity of the proposed recognition method.

In order to attest the contribution of selecting the discriminating patches in the proposed method, we compared the facial expression recognition performance with those without selection and with selection. This comparison concerns not only the recognition rate, but also the size of the feature vector and the time execution. Table 3.3.2 shows this assessment.

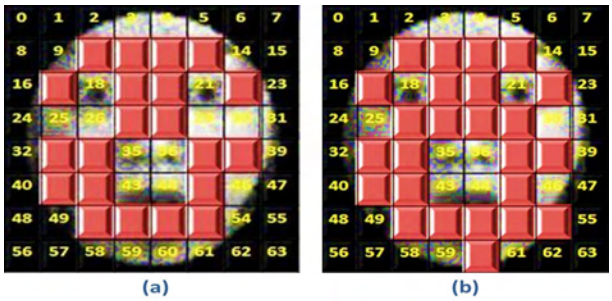


Figure 8: The selected patches using the LBP operator (a) and the DWT operator (b)

Relying on the obtained results, three conclusions are drawn. The first is the contribution of selecting discriminative patches in terms of performance: a gain of 0.47% in facial expression recognition rate. The second is the contribution in terms of space memory: a gain of more than 3 times in the size of the feature vector. The third is the contribution in terms of speed: a gain in time execution of almost 5 time, which is very important for real-time applications.

3.3.3 Third series of experiments

This series of experiments aims to compare the proposed method performance with the most known works in the literature [SO04] [ZZ11] [LBA99b] [ZLSA98]. For fair comparison, we selected the methods which performed their experiments on JAFFE database with a 10-cross-validation evaluation technique. Table 3.3.3 shows this comparative study.

	Without selection	With selection
Number of patches	64	21
Feature vector size	16384	5376
Time execution per image (ms)	19 ms	04 ms
CCR	93.42 %	93.89 %

Table 2: The contribution of patches selection in terms of CCR and time execution

Methods	[LBA99b]	[ZLSA98]	[SO04]	[ZZ11]	The proposed method
CCR	92.00%	90.10%	69.40%	81.59%	93. 89%

Table 3: Comparative study between the proposed method and some previous works on JAFFE database

From Table 3.3.3, the proposed method affords the best recognition rate (93.89%), whereas the highest rate recorded by the studied methods is 92.00%.
Besides the satisfied results in terms of the recognition rate and the required memory space, we have shown through this series of experiments that our method has the advantage of being much faster with a significant gain in the execution time.

4 CONCLUSION

This paper introduces a new method for facial expression recognition using the most discriminant facial patches. These patches were selected automatically using the mutual information technique. Facial feature extraction was performed using the LBP operator applied only on the preselected facial patches. The experimental study showed the improvement while using only salient patches. In fact, we succeed not only to improve facial expression recognition performance but also to speed up the recognition task which is a very important gain for real time applications.
As future work, we intend to experiment our method with more different facial expression databases. Furthermore, we aim to include the temporal information of facial expressions which may provides more accurate classification results.

5 REFERENCES

[BLFM03] M. S. Bartlett, G. Littlewort, I. Fasel, and R. Movellan. Real time face detection and facial expression recognition : Development and application to human computer interaction. In *CVPR Workshop on CVPR for HCI*, 2003.

[CW02] J. T. Chien and C. C. Wu. Discriminant wavelet faces and nearest feature classifiers for face recognition. *Journal of IEEE Trans. Patt. Anal. Mach. Intelligence*, pages 1644–1649, 2002.

[DC99] M. Dailey and G. Cottrell. Pca gabor for expression recognition. *The Journal of UCSD Computer Science and Engineering Technical Report*, 1999.

[DGG06] K. Delac, M. Grgic, and S. Grgic. Independent comparative study of pca ica and lda on the feret data set. *The Journal of Wiley Periodicals*, pages 1121–1137, 2006.

[Ekm72] P. Ekman. Universals and cultural differences in facial expressions of emotions. In *The Nebraska Symposium of Motivation*, 1972.

[FC15] Y. Fang and L. Chang. Multi-instance feature learning based on sparse representation for facial expression recognition. *Journal of MultiMedia Modeling*, pages 224–233, 2015.

[FJJ09] L. W. Feng, L. S. Juan, and W. Y. Jiang. Automatic facial expression recognition based on local binary patterns of local areas. In *WASE International Conference on Information Engineering*, 2009.

[GD03] G. D. Guo and C. R. Dyer. Simultaneous feature selection and classifier training via linear programming: A case study for face expression recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 346–352, 2003.

[Ham06] Z. Hammal. *Segmentation des traits du visage, analyse et reconnaissance d'expressions faciales par le Modele de Croyance Transferable*. PhD thesis, Universite de Joseph Fourier de Grenoble, 2006.

[KCT00] T. Kanade, J. Cohn, and Y. L. Tian. Comprehensive database for facial expression analysis. In *International Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[LBA99a] M. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *Journal of IEEE Transaction Pat-*

tern Analysis Machine Intell, pages 1357–1362, 1999.

[LBA99b] M. J. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *The Journal of IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1357–1362, 1999.

[LBF+04] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. In *IEEE Workshop on Face Processing in Video*, 2004.

[LFCY06] S. Liao, W. Fan, A. C. S. Chung, and D. Y. Yeung. Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features. In *The International Conference on Image Processing*, pages 665–668, 2006.

[LP12] R. Londhe and V. Pawar. Facial expression recognition based on affine moment invariants. *The IJCSI International Journal of Computer Science Issues*, pages 388–392, 2012.

[Mag07] A. F. Magalhaes. *The Psychology of Emotions : The Allure of Human Face*. Pessoa Press, 2007.

[MS00] E. Morales and F. Y. Shih. Wavelet coefficients clustering using morphological operations and pruned quadtrees. *Journal of Pattern Recognition*, pages 1611–1620, 2000.

[OPH96] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Journal of Pattern Recognition*, pages 51–59, 1996.

[PLD05] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency max-relevance and min-redundancy. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1226–1238, 2005.

[SG08] C. Shan and T. Gritti. Learning discriminative lbp-histogram bins for facial expression recognition. In *British Machine Vision Conference*, 2008.

[SGA09] B.L. Sheaffer, J.A. Golden, and P. Averett. Facial expression recognition deficits and faulty learning: Implications for theoretical models and clinical applications. *International Journal of Behavioral Consultation and Therapy*, pages 31–55, 2009.

[SGM05] C. Shan, S. Gong, and P. W. McOwan. Robust facial expression recognition using local binary patterns. In *IEEE International Conference on Image Processing*, pages 370–373, 2005.

[SJD08] P. Sungsoo, S. Jongju, and K. Daijin. Facial expression analysis with facial expression deformation. In *The in Proc. IAPR Int. Conf. Pattern Recog*, pages 1–4, 2008.

[SO04] Y. Shinohara and N. Otsu. Facial expression recognition using fisher weight maps. In *The IEEE Int Conf Automatic Face and Gesture Recognition*, pages 499–504, 2004.

[Soo00] E. Soofi. Principal information theoretic approaches. *Journal of the American Statistical Association*, pages 1349–1353, 2000.

[ST08] C. Shan and T.Gritti. Learning discriminative lbphistogram bins for facial expression recognition. In *British Machine Vision Conference*, 2008.

[Vap98] V. N. Vapnik. Statistical learning theory. *Wiley*, 1998.

[VJ04] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, pages 137–154, 2004.

[VTG+15] M. F. Valstar, T.Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. Fera 2015 - second facial expression recognition and analysis challenge. In *the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8, 2015.

[ZLSA98] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor wavelets-based facial expression recognition using multilayer perceptron. In *Automatic Face and Gesture Recognition*, pages 454–459, 1998.

[ZZ11] X. Zhao and S. Zhang. Facial expression recognition based on local binary patterns and kernel discriminant isomap. *Journal of Sensors*, pages 9573–9588, 2011.

[ZZG04] B. L. Zhang, H. Zhang, and S. Ge. Face recognition by applying wavelet subband representation and kernel associative memory. *Journal of IEEE Trans Neural Networks*, pages 166–177, 2004.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

A new approach to turbid water surface identification for autonomous navigation

Mateus Eugênio Colet, Adriana Braun, Isabel H. Manssour
PUCRS, Faculdade de Informática
Porto Alegre, RS, Brazil
mateus.colet@acad.pucrs.br, adriana.braun@gmail.com, isabel.manssour@pucrs.br

ABSTRACT

Navigation of autonomous vehicles in natural environments based on image processing is certainly a complex problem due to the dynamic characteristics of aquatic surfaces, such as brightness and color saturation. This paper presents a new approach to identify turbid water surfaces based on their optical properties, aiming to allow automatic navigation of autonomous vehicles regarding inspection, mitigation and management of aquatic natural disasters. More specifically, computer vision techniques were employed in conjunction to artificial neural networks (ANNs), in order to build a classifier designed to generate a navigation map that is interpreted by a state machine for decision making. To do so, a study on the use of different features based on color and texture of such turbid surfaces was conducted. In order to compress the extracted information, Principal Component Analysis (PCA) was performed and its results were used as inputs to ANN. The whole developed approach was embedded in an aquatic vehicle, and results and assessments were validated in real environments and different scenarios.

Keywords

Computer Vision, Surface Vehicle, Principal Component Analysis, Artificial Neural Network.

1 INTRODUCTION

With recent technological advances, several areas of knowledge have been benefited from techniques of digital image processing and computer vision. The area of robotics, mainly, stands out by the wide use of computer vision, in order to acquire necessary knowledge for agents from the universe around them. In addition to the use of sensors, computer vision can provide more information to increase and analyse the amount of data that can be supplied [IMM09a]. Navigation in natural environments based on image processing is certainly a complex problem. The main difficulties are the dynamic characteristics that aquatic surfaces can present, due to variation of image features such as brightness and color saturation. Physical factors such as light intensity, shadows, reflections, diffraction and refraction effects also influence the identification process for navigation [IMM09a].

International organizations related to risk reduction show statistics stating that the impact of floods affects over 500 million people, with a cost of $ 50 million annually, and it accounts for the highest number of deaths registered in natural disasters [Kro15a]. The effects of these disasters are even more drastic in developing countries, due to lack of early warning systems, flood control and emergency response infrastructure [SKV11a]. Considering this context, we are interested in developing an approach to assist in the navigation of an autonomous vehicle in a post-disaster environment, both for gathering data and identifying its real dimension. Some solutions for the navigation of surface vehicles using computer vision have already been developed. However, there are still some open problems, as the availability of a method to navigate in turbid water surfaces in adverse environments, which could run in a hardware with computational limitations and could be adapted to several autonomous vehicles.

The main goal of this paper is to present an approach for automatic identification of navigable turbid water surfaces, based on computer vision techniques. We focus on the key subproblem of automatically segmenting turbid aquatic surfaces for autonomous navigation. The result of this process is the generation of a navigation map to guide the direction to be taken by the autonomous vehicle. Seeking to improve accuracy, two ANNs were trained: the first one to recognize turbid water regions without reflection and the second one to identify those regions with reflection [GSW07a, ASN11a]. These ANNs, as well as all the algorithms of this approach, run independently in an embedded hardware. A navigation map is built and, then, a finite state machine guides

the direction to be taken by the autonomous vehicle, which can be a boat, a navigable platform or a smaller device.

The main contributions of the presented approach are:

- Proposal of a method to estimate navigable turbid water surfaces from images captured by a monocular camera positioned on an aquatic vehicle;
- Generation of a navigation map that determines the limits of navigable regions and that can be interpreted by other algorithms for navigation;
- Presentation and development of an algorithm for decision making related to autonomous navigation, based on the generated navigation map;
- Embedding the developed approach in an aquatic vehicle.

The remainder of this paper is organized as follows. Section 2 presents some related works. We briefly describe the developed approach in Section 3. In Section 4 we present some experiments and results. Finally, the last section presents closing comments and future works.

2 RELATED WORK

Some solutions of locomotion for surface vehicles using computer vision have already been developed [SMB12a, SMH04a, GSW07a, HS11a, ASN11a, RMB11a]. A detailed description of consequences, influences and variations in variable values, as well as the challenges that effect the ability to detect water surfaces by optical means, is presented by Iqbal et al. [IMM09a]. They focus on difficulties involved in the detection of water bodies, along with state of the art techniques that deal with this topic. Andrew et al. [CAN11a] also emphasize that aquatic environments present several problems, such as the reflection of other objects on the water surface, currents and waves that distort the aspect of water, or the presence of debris or sediment, which changes the color of water or causes movement on its surface.

According to Huntsberger et al. [HAH11a] and Yao et al. [YXL07a], a vehicle equipped with a water detector based on computer vision has a higher probability to navigate safely and efficiently. This is particularly emphasized when the vehicle is in an unknown environment. The image acquisition in this case can be done by a set of cameras [SMH04a] or just a monocular camera [SMB12a, CAN11a, YRC11a].

For example, Santana et.al. [SMB12a] propose a model for water detection with segmentation guided by dynamic texture recognition. From an input video, they defined that the region of water has a signature, based on the measure of entropy over the trajectories obtained from optical flow trackers. In order to classify regions with a higher degree of reliability in surfaces of little ripple, a segmentation method based on appearance is applied. Then, every image is labeled in segments with water if they cover a certain percentage of pixels classified as water by the method based on entropy and texture. It presented a good true positive rate; however, this model does not adapt to mobile cameras due to its constant movement. According to the authors, tracking stabilization techniques would help in reducing the inertial optical flow induced by camera moving.

Rankin and Matthies [RM10a] proposed the detection of water bodies and ponds through the behavior modeling of these surfaces. They used intensity data based on the variation of color spaces RGB and HSB to estimate the contribution of the reflection coefficient, considering the reflection surface and a combination of other factors such as saturation and brightness. According to the authors, the developed method for detecting water bodies in open areas proved to be sensitive to any reflection, both vegetation and objects on the aquatic surface. One way to deal with possible exceptions could make this method more robust and less limited.

Other works [GSW07a, ASN11a, HS11a] use a robust descriptor with the analysis and combination of features to build a classifier with supervised learning. According to the authors, the use of a set of training data allows to build a good classifier to distinguish water surfaces, since natural environments suffer variations, as physical factors.

Gong et al. [GSW07a] present a two-stage algorithm to find the margin between water and land. Images are collected and classified into two types: Reflection-identifiable and Reflection-unidentifiable. After, the images are segmented into smaller regions based on their color and uniformity, which are classified into areas of land or water according to features such as symmetry and brightness. Then, the algorithm traces a border to separate water from land regions by means of a classifier using an adaptive threshold segmentation. Frames with 320 x 240 pixels with reflection-identifiable processing take 2 seconds to be processed, and frames with reflection-unidentifiable take 27 seconds. Besides being a computationally expensive algorithm, it also seems hard to be implemented in autonomous video capture application.

Achar et al. [ASN11a] propose a self-supervised method to segment images into "sky", "river" and "shore" regions. It uses assumptions about river scene structure to learn about appearance models based on features as color, texture and image location, considering the horizon line to automatically specify the correlation among features. It extracts features of color spaces RGB, Lab and HSV individually and in various

combinations to train the classifier. Thus, it allows to label each part of the image with the probability of being water. Each labeled region is used to train a support vector machine (SVM) model generating the output for each image segment. This method presents good results, but each frame of 640 *x* 360 pixels takes around 2.32 seconds to be processed in a high-performance computer. Thus, it is difficult to embed it in medium and small vehicles.

Considering the methods described in several works [RM10a, ASN11a, IMM09a], we have adopted the use of several features (see Section 3.2) for the development of our approach. This is because the aquatic surface not only changes its optical property such as saturation and brightness, but it is also not uniform, causing color variation. Some techniques are robust to distinguish, segment and identify aquatic surfaces based on color analysis, and by using several color spaces, it states that a color descriptor associated with a vector of features becomes robust using statistical measurements to form classifiers [RM10a, ASN11a, HS11a, GSW07a].

3 APPROACH DESCRIPTION

This section presents the proposed approach for the automatic identification of navigable turbid water surfaces and automatic navigation of aquatic vehicles. It starts with an overview of the developed methodology, followed by an explanation about each implemented step.

3.1 Methodology Overview

The developed approach has several steps, as presented in Figure 1. Initially, sequences of images are collected by a monocular camera coupled to the prototype of the autonomous aquatic vehicle shown in Figure 11-(b). Then, the first step consists in the subdivision of each input frame I into blocks of $r \times s$ pixels. The values of r and s should be set to ensure good computational performance and classification granularity. In our experiments, we set $r = s = 10$ pixels, since our input frames have 320×240 pixels. Thereafter, for each block B, a set of 32 colors and texture features is extracted (see Section 3.2). We standardized these features and changed the coordinates of z-scores, by projecting them into the subspace of k principal components obtained through Principal Component Analysis (PCA) for the training phase described in Section 3.3, hence reducing data dimensionality. The values obtained are submitted to the classifiers, modeled as multilayer perceptron Artificial Neural Networks (ANN). As output for the ANNs, each image block B is classified as a "navigable" or "non navigable" region, independently. This procedure allows us to classify each block in different threads, which increases computational performance. Once all image blocks have been classified, we built a navigability map for each frame. This map is then submitted to a Finite State Machine (FSM), that interprets it and defines the actions to be performed by the vehicle. The following sections describe this methodology.

3.2 Extraction of image features

Each image block B is processed individually as follows: Firstly, we convert it to HSV and YUV color spaces, keeping the original RGB block image; afterwards, it is split into 8 color channels (red, green, blue, hue, saturation, value, luminance and chrominance). Then, we compute a series of statistics for each one, as described below.

Initially, we computed the normalized histogram of intensities for each channel c. Hereafter, these histograms will be denoted as H_c, with $c \in$ { *Red, Green, Blue, Hue, Saturation, Value, Y(luminance), U(chrominance)* }. The value of element h_{ci} from the histogram H_c is given by:

$$h_{ci} = \frac{n_i}{n}, \tag{1}$$

where $i \in [0, M]$, n_i is the number of pixels with intensity i in each channel c of a given image block, $n = r \times s$ and M is the maximum intensity value of the color channel, i.e., $M = 255$ considering a color depth of 8 bits per pixel.

Given the eight normalized histograms H_c, the following statistics are computed:

- **Average:**

$$\upsilon_c = \sum_{i=0}^{M-1} i * h_{ci}. \tag{2}$$

- **Entropy:**

$$E_c = -\sum_{i=0}^{M-1} h_{ci} \log_2 h_{ci}. \tag{3}$$

- **Variance:**

$$\sigma_c^2 = \sum_{i=0}^{M-1} (i - \upsilon_c)^2 * h_{ci}, \tag{4}$$

- **Energy:**

$$\varepsilon_c = \sum_{i=0}^{M-1} (h_{ci})^2. \tag{5}$$

After all these statistical measurements have been computed, 32 features per image block were generated (average, entropy, variance and energy of the 8 color channels). Next sections explain how these features are used to train ANNs and, subsequently, as inputs for turbid water recognition.

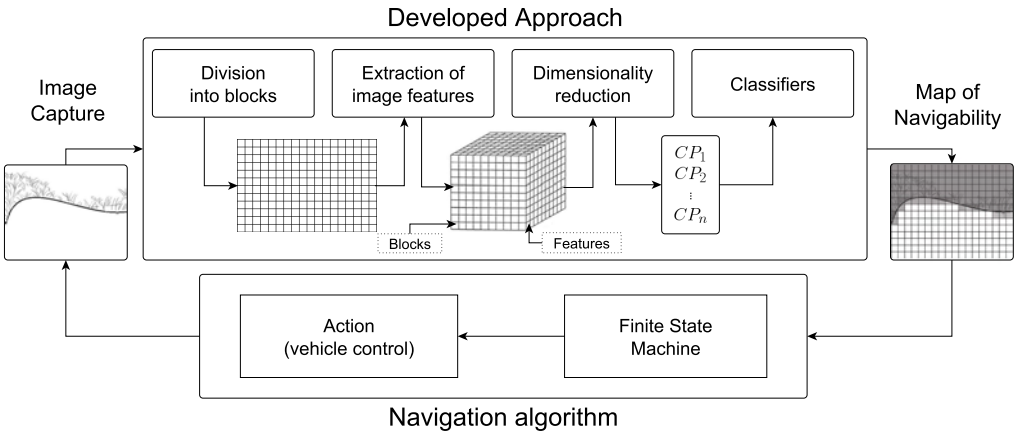


Figure 1: Diagram showing the steps of the proposed methodology.

3.3 Preprocessing and Training

In order to accomplish the proposed goals, we used a supervised approach, i.e., we trained our classifiers using labelled features. Thus, for our experiment, we used a video made in the scenario presented in Figure 10 as a training environment. We selected a set of 15 frames randomly chosen to cover different conditions of luminosity and water turbidity. The acquisition was performed through a monocular camera attached to the prototype vehicle described in Section 4.3. These images were then divided into blocks as previously explained, and the 32 features were extracted.

Next, we performed the manual annotation of image blocks. To this end, an interactive tool was built, in which users were asked to paint in green all navigable regions from the input images, through mouse interactions. Users were supposed to paint disjoint regions, according to the presence of water or not. Figure 2 presents two annotated frames, where blocks marked in red are "not navigable" and the blocks marked in green are "navigable".

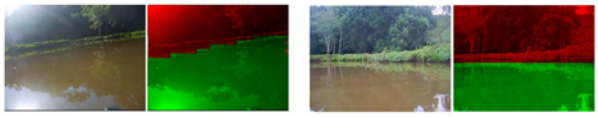


Figure 2: Annotation process of training frames.

A common procedure to avoid data over-fitting and to increase the generality and convergence speed of pattern recognition methods is to employ a dimensionality reduction technique [Bis06a]. We chose to apply Principal Component Analysis of features to accomplish this goal.

First of all, we performed the standardization of the features from the samples. For this, for each image block, the 32 previously described features were computed. We normalized every feature f_j with respect to their range of values, as follows:

$$\hat{f}_j = \frac{f_j - f_{min}}{f_{max} - f_{min}} \ , \tag{6}$$

where $\hat{f}_j$ is the normalized value of each feature, f_j is the original value of the feature, $j = 1, 2, ..., 32$, f_{min} is the minimum value of feature j, and f_{max} is the maximum value of feature j, considering all training blocks. Given the normalized values, we computed, for each feature j, the average μ_j and the standard deviation σ_j, considering all samples. Then, each extracted feature $\hat{f}_j$ was standardized, according to the equation:

$$z_j = \frac{\hat{f}_j - \mu_j}{\sigma_j}. \tag{7}$$

The z-scores of the features computed through Equation 7 of every training block were finally submitted to PCA. The use of PCA as a preprocessing step for a machine learning method can accelerate its convergence, since it allows dimensionality reduction and the correlation among features [YZL06a]. Figure 3 shows the labeled z-scores projected in the sub-space defined by the three principal components, achieved through PCA. We can notice a visible separation of the blocks classified as navigable (red) from the non-navigable ones (blue). We can also observe that this separation is non-linear. Due to this fact, an artificial neural network was employed as a classifier.

The values of μ_j, σ_j and the matrix of the sorted eigenvector from the covariance matrix M obtained in PCA are stored to be used to compute features from images acquired during the experiments conducted in real environments described in Section 4.3. The eigenvectors are sorted according to crescent order of eigenvalues.

We defined the ideal dimension of principal components based on the work by Ian [Jol02a] to minimize the complexity subject to a limit on the fidelity of the problem. According to the author, the set of components

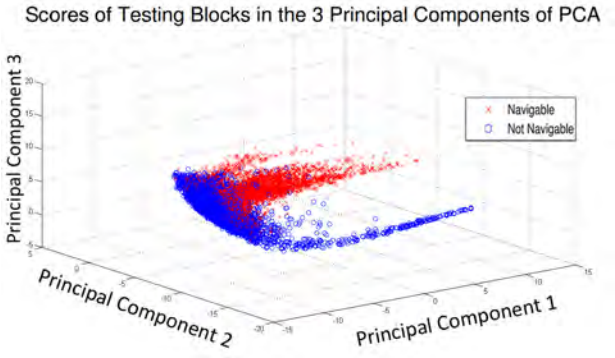


Figure 3: Standardized features from training blocks projected into the subspace defined by the three principal components from PCA.

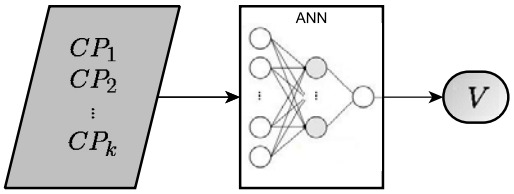
for analysis of values is above a threshold $\hat{r}$ eigenvalues ≥ 1. In our experiments we verified that $81,25\%$ of data variability are incorporated by projecting standardized features into the subspace of the six principal components. Due to this fact, the data coordinates in the subspace of dimension $k = 6$ are then used as input for training the ANN. Thus, the data dimensionality was reduced to $k = 6$. The values μ_j and σ_j are kept and used to standardize the features extracted from new image blocks that must be classified when the vehicle is operational. Next section details the architecture, training and performance of ANNs.

3.4 Classifier

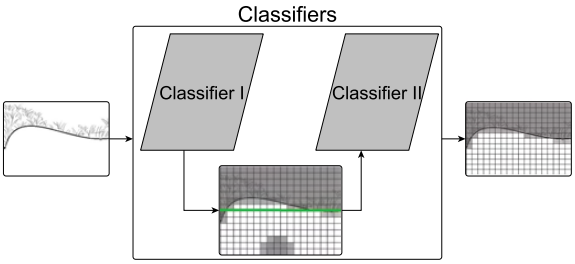
This subsection presents the two ANN classifiers developed for the turbid water surface identification. The purpose of ANN classifiers is to determine if an image block corresponds or not to a navigable surface. The classifier can be defined as follows: B is a block of an image to be classified and $CP = \{ CP_1, CP_2, \cdots CP_k \}$ is the set of coordinates of the extracted features in the subspace of k principal components, computed as explained in the previous section. Thus, the ANN classifier receives the CP's scores as input and returns a value $V \in [0,1]$. The smaller the value of V, less likely it is for a block to correspond to a navigable surface; whereas the greater the value of V, the greater the probability of being a block that corresponds to a navigable surface. We trained two ANNs: the first one described in Section 3.4.1, aims to recognize turbid water surfaces, and the second one aims to recognize regions of reflection on these surfaces, as explained in Section 3.4.2. Figure 4 shows the scheme for the classifiers' modelling (a) and their final architecture (b).

3.4.1 Navigable Surface Identification

In order to solve the problem stated in this approach, we adopted a 3-layer Multilayer Perceptron topology for ANNs [Bha10a]. The input layer has k neurons since



(a) Classifier modeling



(b) Architecture of classifiers

Figure 4: (a) shows how each classifier is built and (b) shows the architecture of our classifiers.

the input data are the set of scores of the training image blocks (in our case, $k = 6$). The intermediate layer has $\frac{k}{2}$ neurons. The output layer has only one neuron since the output is a scalar value $V \in [0,1]$. Figure 4-(a) shows the modeled classifier. If $V < 0.5$, the output means that the image block belongs to a non navigable region; if $V >= 0.5$, that block will be considered as navigable. Intermediate values indicate a low confidence in the classification. Figure 5 shows the values of V for image blocks from the input image on the left, mapped to greyscale images.

We use the resilient propagation algorithm for training our multilayer feedforward network [KNS99a]. According to Svozil et al. [SKP97a], it increases the resolution capability for non-linear problems and ANN becomes very robust, i.e., their performance degrades gracefully in the presence of increasing amounts of noise. In this algorithm, synaptic weights of the network are adjusted according to signal error propagation [Hay98a]. In order to plan an assessment of the convergence of ANNs in the training phase, we used the method developed by Shinzato at al. [SGOW12a]. This method assigns a weight to the classification error for a given ANN, by computing a score S. For the sake of exemplification in this method, a greater weight is assigned to an ANN output with error of 0.1 than an output with error of 0.2. Through this weighted score, there is a tendency to "reward" ANNs with fewer large errors or several small errors and "punish" the other ones. Equation 8 shows how to calculate the score:

$$S = \frac{\frac{1}{N.p(0)} \left(\sum_{i=0}^{hmax} h(i).p(i) \right) + 1}{2.0}, \tag{8}$$

where N is the number of classes ($N = 2$), $h(i)$ is the number of errors ranging from $\frac{i}{hmax}$ to $\frac{i+1}{hmax}$ and $hmax$ is the number of intervals to be considered for discretization in the error counting process. The value of $hmax$ determines the precision for interpretation of the output from the ANN. In other words, e.g., if $hmax = 10$, then the output that has a real value ranging from 0 to 1 is divided into 10 intervals of errors: an error interval for values between 0.0 and 0.1, other error interval for values between 0.1 and 0.2, and so on. After the training phase, the network is executed for each block.

The training of ANNs is repeated until the convergence is reached. In the proposed implementation, the stop criteria adopted is $S = 95\%$ or a limit of 5.000 epochs. Our first ANN reached 95.52% in 1.730 epochs, and the second ANN 95.29% in 460 epochs. After this process, we kept the best ANN, to be used in real time for image blocks' classification.

3.4.2 *Reflection Zone Identification*

In our experiments, we learnt from the classification results of the ANN described in the previous section that a high number of false negatives occur in regions with high reflectance on the water surface. In order to address this problem and enhance performance, a second ANN was trained. The goal of this second ANN is to correctly classify blocks belonging to reflection zones on the water surface as navigable. The topology of this second ANN is the same as described in Section 3.4. The input for the training step consists of features extracted from image blocks manually classified as "reflection zone" (therefore, "navigable") and "non reflection zone". These features are extracted following the same procedures described on Section 3.3.

Due to similarities of reflection zone features and other non navigable zone features, we only applied this second ANN to image blocks below an automatically computed horizon line $L1$ (shown in Figure 5). This line is defined by the upper row of blocks that have at least 25% of classification as "navigable" by the first ANN. Only features extracted from blocks classified as "non navigable" by the first ANN and below the horizon line are submitted to the second ANN. Figure 4-b illustrates this flow. Images in Figure 5 exemplify inputs and outputs from both ANNs. On the left side images, one can note reflection zones on the turbid water surface. The output of the classification computed by the first ANN is shown in the central images, where values of V are mapped to greyscale values (brighter blocks indicate navigable regions). The red ellipses in Figure 5 indicate false negative zones due to reflection and $L1$ indicates the horizon line. Then, the images with blocks classified as "non navigable" by the first ANN that are below $L1$ are submitted to the second ANN.

Figure 5: On the left are the two input images; the images of the middle show the results from the classification of the first ANN, with reflection zones marked by the ellipses; the images on the right side show the result with the combination of both ANN classifiers.

3.4.3 *Classifying New Images*

The model described in previous sections was embedded in an aquatic vehicle, as a prototype. More details on this prototype can be found in Section 4.3. Once the vehicle is on the water surface, new images are acquired by the coupled camera. These images are converted to HSV and YUV space colors, split into 8 color channels, divided into blocks and, for each block, statistics defined in Equations 2 to 5 are computed. This features are then normalised and standardized, according to Equations 6 and 7, which lead us to z-score values.

Given the set of z-scores of new image blocks, we must project them into coordinates of the PCA space. To this end, we used the autovector matrix M for the change of basis of the extracted features:

$$PC = M.Z. \tag{9}$$

where M is the matrix of sorted eigenvectors obtained by PCA, and Z is the vector of normalized and standardized features extracted from each block.

The scores of k's principal components corresponding to each block are then submitted to the first ANN. The horizon line $L1$ is then determined. Blocks with value $V < 0.5$ assigned by the first ANN below the horizon line are submitted to the second ANN. The output of these procedures is a matrix, whose elements correspond to an image block. From now on, this matrix will be addressed as map of navigability. This map will guide the decision-making about the direction the vehicle must follow. Next section explains how the decision-making process was implemented.

3.5 Navigation algorithm

Given the navigation map composed by the output from blocks' classification, a decision making process must be employed to guide the navigation of the aquatic vehicle. This process begins with the subdivision of the navigation map into four regions, as shown in Figure 6-(a). For each navigation map, the regions $SP1$, $SP2$, $SP3$ and $SP4$ are defined by lines $L1$, $L2$ and $L3$. $L1$ is the horizon line that also appears in in Figure 5.

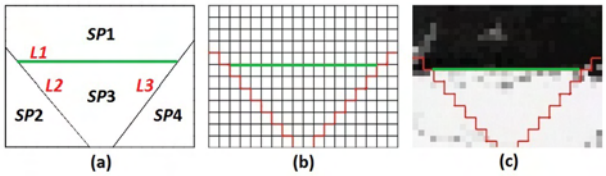


Figure 6: (a) Definition of areas for decision making; (b) search of classifications based on predefined areas; (c) combination of predefined areas with the navigation map.

Lines $L2$ and $L3$ of Figure 6(a) are defined, respectively, according to:

$$\begin{aligned} y_2 &= \tfrac{1}{3}nr + x_2, \\ y_3 &= \tfrac{1}{3}nr + nc - x_3, \end{aligned} \tag{10}$$

where nr is the number of rows of blocks, nc is the number of columns of blocks, y_2 and y_3 are the rows, x_2 and x_3 are the columns of lines $L2$ and $L3$, respectively. The origin is in the upper left corner of the navigation map and y is oriented top down.

Once the areas are delimited, an FSM defines if the vehicle must remain in the same direction, turn left, turn right or stop. First of all, we computed the number of blocks classified as "navigable" in each region. The decision process can be summarised as follows: (1) if the row of the horizon line $L1$ is higher than $\frac{2}{3}.nr$, the vehicle should stop. This occurs mainly when there is few or no navigable blocks ahead of the vehicle; (2)if the number of blocks classified as "navigable" in $SP2$ is higher than in $SP3$ and $SP4$, the vehicle should turn left; (3) if the number of blocks classified as "navigable" in $SP3$ is higher than in $SP2$ and $SP4$, the vehicle should keep forward; (4) if the number of blocks classified as "navigable" in $SP4$ is higher than in $SP2$ and $SP3$, the vehicle should turn right.

Figure 7 illustrates this FSM, with the diagram of actions to be taken according to analysis carried out on the navigation map and the predefined areas.

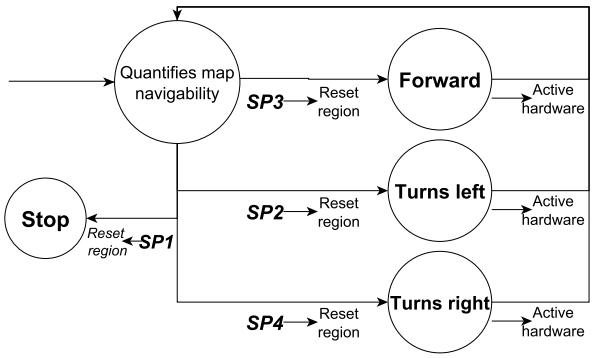


Figure 7: Decision diagram for FSM actions.

The approach to decision making is a proof of concept, developed to ensure fast performance when embedded in the aquatic vehicle. We use FSM because we can easily describe a sequence of states considering different contexts for each input image. Then, it is easy to change from one state to another, defining a specific action to be taken for each state. Next section presents and discusses the results achieved by our approach.

4 RESULTS AND DISCUSSION

This section presents some obtained results, aiming to validate the presented approach. Subsection 4.1 presents the scenarios where the images were collected. The performance metrics evaluated and the results of the tests in real environments are presented in subsection 4.2. In subsection 4.3 we describe our prototype.

4.1 Images and environment

We chose three different environments for extracting the images used to evaluate the developed approach. All images were collected from these environments under different timetables and after a heavy period of rain, in order to achieve the characteristic of turbid water surface. Thus, we tried to approximate as close as possible to the conditions of a real situation where an autonomous vehicle can assist navigation in a post-disaster environment. Figure11-(c) shows our prototype in action, and some of these frames of each evaluated scenario are presented in Figure 8. Each scenario with its peculiarities will be further described.

4.1.1 Scenarios' description

Scenario I corresponds to a rural environment, made up entirely of vegetation, with many trees, rocks and grass on the slope. Figure 8-(a) exemplifies some frames of this scenario. It is possible to notice on these images that the aquatic surface presents large incidence of reflection of the sky, changing the optical property of the turbid water surface.

Scenario II is also a rural environment, but it presents less vegetation and some houses, some of which even working as a form of obstacle to the boat. Figure 8-(b) presents some frames of scenario II. In these images, it is possible to see that there was little incidence of sky reflection on the water surface, showing a subtle reflection of vegetation and houses.

Scenario III corresponds to an urban environment, depicting a real situation of natural disaster. This environment is more complex, since it presents heterogeneous situations. As shown in Figure 8-(c), the images extracted from this scenario can contain, for example, people, cars, animals, and buildings.

4.2 Approach evaluation

Considering the ROC (Receiver Operating Characteristics) analysis [Faw06a], the evaluation was performed

(a) Frames of scenario I

(b) Frames of scenario II

(c) Frames of scenario III

Figure 8: Some examples of frames illustrating each environment used to evaluate the developed approach.

in terms of *accuracy*, *sensitivity* and *precision*, as defined in Equations 11, 12, and 13, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{11}$$

$$Sensitivity = \frac{TP}{TP+FN}, \tag{12}$$

$$Precision = \frac{TP}{TP+FP}, \tag{13}$$

where TP, TN, FP, and FN refer to True Positive, True Negative, False Positive and False Negative, respectively.

We consider that the proposed approach had a satisfactory accuracy rate in our experiments (Figure 9). It achieved an average accuracy of 95.85% with standard error of 0.924 for scenario I, an average accuracy of 93.35% with standard error of 0.882 for scenario II, and an average accuracy of 91.21% with standard error of 0.980 for scenario III. Figure 10 shows the results for some random frames classified in each scenario.

By analyzing the generated average values for each scenario, it is possible to verify better results of the evaluated metrics for the first scenario. One reason for this may be due to the fact that some images acquired in this scenario were used for training the ANN. Scenario II is quite similar to the first one used for training. Thus,

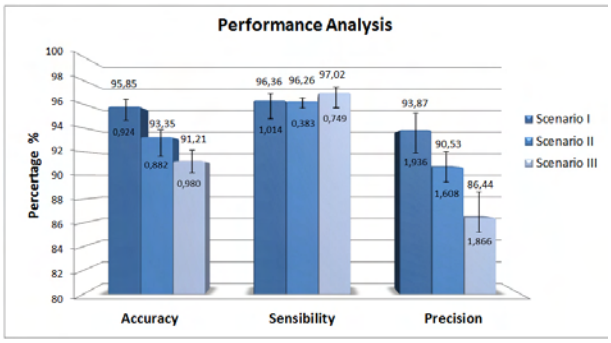


Figure 9: Evaluation results for each scenario.

although it is an unknown scenario, a good result was obtained with an average of 90% among the evaluated metrics. Scenario III corresponds to an adverse environment with a lot of diversity, such as people, houses, cars and objects floating on the water surface. Even so, the approach proved to be efficient, since it has a good sensitivity evaluation, which demonstrates a good performance in identifying the surface with a high rate of true positive values. On the other hand, lower values for precision are due to high rate of false positive values.

4.3 Embedded approach

In order to evaluate the developed approach for autonomous navigation in a real environment, it was embedded in an aquatic vehicle. We build and develop our approach using the programming language C, with support of OpenCV library, OpenMP for multiprocessing programming, and Fast Artificial Neural Network Library (FANN), a free library that implements an ANN multilayer in language C [Nis05a]. The hardware used was a Raspberry Pi board (RPI) model 2 and a Raspberry camera. Figure 11-(b) shows the prototype of the aquatic vehicle with the RPI board and camera connected. Its advantage is the processing totally made on the boat, without the need of having communication or sending commands through an external computer.

Results achieved with the RPI 2 board were: 46% of processor usage, 308.9 MB of memory for execution and 2.5 frames per second (FPS). Figure 11-(a) shows our approach running on the operating system RPI 2. Analyzing the performance of obtained results and considering the usual speed of aquatic vehicles it's possible to say that 2.5 FPS is an acceptable performance.

We used our prototype to evaluate the navigation algorithm, which is based on the generated navigation map. For this evaluation, we collected 48 frames of the described scenarios, with twelve frames for each possible action command defined in our FSM (four for each scenario). Then, we analysed each frame to define the best action or the expected command considering the aquatic surface and its obstacles, and we compared them with the executed command. Table1 presents the

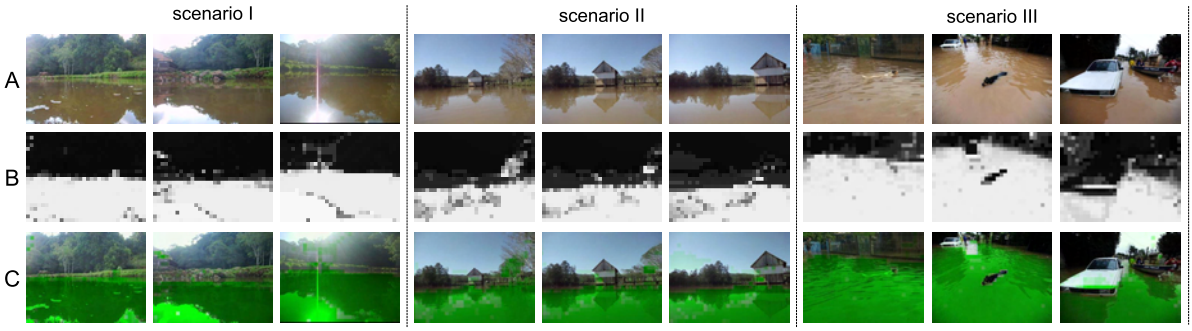
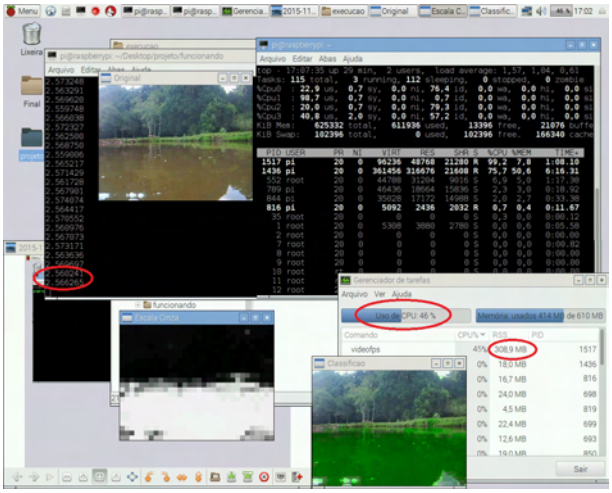


Figure 10: Results obtained for each scenario: (a) input frame; (b) map of generated navigability, and (c) overlay to indicate the navigable region.



(a)

(b) (c)

Figure 11: (a) Performance evaluation of the approach on the RPI 2 board; (b) Prototype built for approach evaluation; (c) Prototype running our approach to collect obtained results in a real environment.

expected commands for these selected frames and the executed commands by our algorithm.

For the obtained average of 66.25%, we considered only the expected values as correct, even though other commands could also be suitable. The low value for the "stop" command is because the vehicle was programmed to stop just when there were few or no navigable blocks ahead of it.

5 CONCLUSION

In this work we proposed an approach for automatic identification of navigable turbid water surfaces, based on computer vision techniques. Artificial neural networks (ANNs) were also used to build a classifier designed to generate a navigation map, and principal component analysis (PCA) was performed to compress the extracted information used as input to ANN.

Table 1: Comparison of expected and executed commands by the developed FSM.

Set of commands defined	Expected Commands	Commands Executed
Forward	12	8
Turn right	12	9
Turn left	12	10
Stop	12	5
Hit average movement	66,25%	

The proposed approach was quantitatively evaluated using a dataset containing images extracted from three different scenarios. Experimental results indicated that the approach effectively identified navigable region achieving between 91.21% and 95.85% of accuracy. For testing and evaluation of our approach, we built a prototype used in three real environments in order to demonstrate the adaptability and viability of our approach to autonomy of aquatic vehicles. Thus, we believe it can be used to assist navigation of an autonomous vehicle in a post-disaster environment.

For future work we intend to use pre and post-processing techniques in the navigability map, mainly to improve false positive results. We would also like to improve our navigation algorithm, in order to develop better search directives on the navigability map and, consequently, execute more precise commands in our FSM. Furthermore, we also want to use other sensors such as laser or distance sensors to increase the capacity of performance in navigation.

6 ACKNOWLEDGMENTS

The authors would like to thank Brazilian agency CAPES for the financial support. This work was also partially supported by PUCRS.

7 REFERENCES

[ASN11a] Achar, Supreeth and Sankaran, B. and Nuske, Stephen. Self-supervised segmentation of river scenes. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 6227-6232. IEEE, 2011.

[Bha10a] R. Bhati. Face recognition system using multi layer feed forward neural networks and principal component analysis with variable learning rate. In Communication Control and Computing Technologies (ICCCCT), IEEE International Conference on, pages 719-724, 2010.

[Bis06a] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[CAN11a] Andrew Chambers, Supreeth Achar, Stephen Nuske, Jorn Rehder, Bernd Kitt, Lyle Chamberlain, Justin Haines, Sebastian Scherer, and Sanjiv Singh. Perception for a river mapping robot. In Intelligent Robots and Systems (IROS), International Conference on, pages 227-234. IEEE, 2011.

[Faw06a] Tom Fawcett. An introduction to roc analysis. Pattern recognition letters, 27(8):861-874, 2006.

[GSW07a] Xiaojin Gong, Anbumani Subramanian, and Christopher L Wyatt. A two-stage algorithm for shoreline detection. In Applications of Computer Vision. WACV. IEEE, pages 40-40. IEEE, 2007.

[HAH11a] Terry Huntsberger, Hrand Aghazarian and Andrew Howard. Stereo vision-based navigation for autonomous surface vessels. Journal of Field Robotics, 28(1):3-18, 2011.

[Hay98a] Simon Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[HS11a] Hordur Kristinn Heidarsson and G Sukhatme. Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In Intelligent Robots and Systems (IROS), International Conference on, pages 3160-3165. 2011.

[IMM09a] Mohammad Iqbal, Olivier Morel, and Fabrice Meriaudeau. A survey on outdoor water hazard detection. International Conference on Information Communication Technology and Systems, pages 33-39, 2009.

[Jol02a] Ian Jolliffe. Principal component analysis. Wiley Online Library, 2002.

[KNS99a] K. Keeni, K. Nakayama, and H. Shimodaira. A training scheme for pattern classification using multi-layer feed-forward neural networks. In Computational Intelligence and Multimedia Applications. ICCIMA. Proceedings. Third International Conference on, pages 307-311, 1999.

[Kro15a] Wolfgang Kron. Flood disasters a global perspective. Water Policy, 17(S1):6-24, 2015.

[Nis05a] Steffen Nissen. Neural networks made simple. Software 2.0, 2:14-19, 2005.

[RM10a] Arturo Rankin and Larry Matthies. Daytime water detection based on color variation. Intelligent Robots and Systems (IROS), IEEE International Conference on, pages 215-221. IEEE, 2010.

[RMB11a] Arturo L Rankin, Larry H Matthies, and Paolo Bellutta. Daytime water detection based on sky reflections. In Robotics and Automation (ICRA), IEEE International Conference on, pages 5329-5336. IEEE, 2011.

[SGOW12a] P.Y. Shinzato, V. Grassi, F.S. Osorio, and D.F. Wolf. Fast visual road recognition and horizon detection using multiple artificial neural networks. In Intelligent Vehicles Symposium (IV), IEEE, pages 1090-1095. IEEE, 2012.

[SKP97a] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems, 39(1):43-62, 1997.

[SKV11a] Paul Scerri, Balajee Kannan, Pras Velagapudi and Kate Macarthur. Flood disaster mitigation: A real-world challenge problem for multi-agent unmanned surface vehicles. In Advanced Agent Technology,pages 252-269. Springer, 2011.

[SMB12a] Pedro Santana, Ricardo Mendonça, and José Barata. Water detection with segmentation guided dynamic texture recognition. In Robotics and Biomimetics (ROBIO), IEEE International Conference on, pages 1836-1841. IEEE, 2012.

[SMH04a] Franklin D Snyder, Daniel D Morris, Paul H Haley, Robert T Collins, and Andrea M Okerholm. Autonomous river navigation. In Optics East, pages 221-232. International Society for Optics and Photonics, 2004.

[YRC11a] Junho Yang, Dushyant Rao, S Chung, and Seth Hutchinson. Monocular vision based navigation in GPS-denied riverine environments. In Proceedings of the AIAA Infotech Aerospace Conference, St. Louis, MO, 2011.

[YXL07a] Tuozhong Yao, Zhiyu Xiang, Jilin Liu, and Dong Xu. Multi-feature fusion based outdoor water hazards detection. Mechatronics and Automation International Conference ICMA, pages 652-656. IEEE, 2007.

[YZL06a] Jun Yan, Benyu Zhang, Ning Liu, and Shuicheng Yan. Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing. IEEE Transactions on Knowledge and Data Engineering, 18(3):320-333, March 2006.

An unsupervised 3D mesh segmentation based on HMRF-EM algorithm

Sabra Mabrouk
CRISTAL Laboratory, ENSI, La Manouba University
Campus Universitaire de la Manouba, Tunisia, 2010, Manouba
sabra.mab@gmail.com

Faten Chaieb
CRISTAL Laboratory, ENSI, La Manouba University
Campus Universitaire de la Manouba, Tunisia, 2010, Manouba
faten.chaieb@ensi.rnu.tn

Faouzi Ghorbel
CRISTAL Laboratory, ENSI, La Manouba University
Campus Universitaire de la Manouba, Tunisia, 2010, Manouba
faouzi.ghorbel@ensi.rnu.tn

ABSTRACT

We propose a new 3D mesh segmentation method based on the HMRF-EM framework. The clustering method relies on the curvature attribute and considers the spatial information encoded by the mutual influences of neighboring mesh elements. A region growing process is then carried out in order to extract connected regions followed by a merging procedure. The purpose of this latter process is to only preserve meaningful regions. Experiments conducted on different meshes are encouraging and show that the proposed method gives satisfying results compared with classical statistical ones such as kmeans and EM algorithms.

Keywords

HMRF-EM algorithm, region growing, region merging, mesh segmentation.

1 INTRODUCTION

3D mesh segmentation has been an important 3D shape analysis topic, essential for a wide range of applications such as part-based shape recognition or retrieval, Sketch-based Shape Retrieval [Cha15], texture mapping, reverse engineering applications that deals with CAD models and component-shape based synthesis that provides new models by combinations of parts from existing models [Kal12].

3D mesh segmentation consists in partionning the mesh into disjoint sub-meshes according to some specific criteria. Many segmentation algorithms have been proposed in the litterature. They could be classified into part-type ones that decomposes the mesh into semantic and meaningful part and patch-type ones based on the mesh geometry attributes such as curvatures, convexity, roughness, etc.

Many 3D mesh segmentation methods have been proposed such as clustering ones, region-growing ones and spectral methods [Sha08]. In this work we focus on clustering methods that aim to associate an appropriate cluster label to each mesh element according to some attribute values. Most of them are 3D extensions of well known 2D classification algorithms [Sha08, Lav08, Tsu14]. Curvature based descriptors are widely used attribute since curvature is a very significant criterion that describes the shape structure variation. However, clustering methods based on curvature attribute generate many isolated fragments as curvature is very sensitive to noise. So a post-treatment step is always needed to deal with such problem.

In [Lav08], a clustering method based on the Markov Random Fields (MRF) schema has been proposed. Authors initialise their proposed method by a K-means algorithm, the resulting labeled mesh is then median filtered and used for the prior and observation parameters estimation. The simulated annealing is subsequently applied for the resolution of the maximun a posteriori estimate (MAP) which is known to be a very time consuming algorithm. It's important to notice that few work dealt with the MRF extension to 3D mesh processing [And07, Wil04, Lav08].

In this work, we propose a new 3D mesh segmentation method based on the HMRF-EM clustering framework [Zan01]. This framework incorporates the HMRF model with the Estimation-Maximization (EM) algorithm. Unlike [Lav08], the iterated conditional mode is adopted for the optimization step that seeks the MAP estimate and for the prior model estimation an adaptive weighted cost function is also defined based on the dihedral angles of neighboring faces. This method

takes into account both spatial and attribute information which yields to be robust to noisy data. In fact, the mesh geometry is encoded through the mutual influences of their neighboring sites. A post treatment based on a region growing method is then carried out in order to generate only connected components.

This reminder of the paper is organised as follows : in section 2, we describe the proposed method overview while section 3 details the HMRF-EM framework adapted for the 3D mesh dual graph. Section 4 deals with the post treatment that aims to extract connected regions from the resulting labled mesh. Finaly, some experiments and results on different meshes are shown in section 5.

2 METHOD OVERVIEW

Figure 1 shows the main steps of the proposed mesh segmentation method. First, the curvature attribute of the input triangular mesh, denoted by $\mathscr{M}(\mathrm{V},\mathrm{F})$ where V is the set of vertices and F the set of triangles, is computed. Then, we carry out a facet-based clustering algorithm that combines HMRF model with EM algorithm. To deal with facet-based clustering we consider the dual graph $\mathscr{M}^*$ of $\mathscr{M}$ that is defined as follows : Each vertex of the dual graph corresponds to a triangle of $\mathscr{M}$ and two vertices of $\mathscr{M}^*$ are neighbors if and only if their corresponding triangles in $\mathscr{M}$ share an edge. The curvature attribute associated to each facet gravity center is the mean curvature values computed on their vertices. It's important to notice that the proposed method could deal with many others attributes rather than curvature ones. Finally a connected region extraction step is performed in order to eliminate isolated parts.

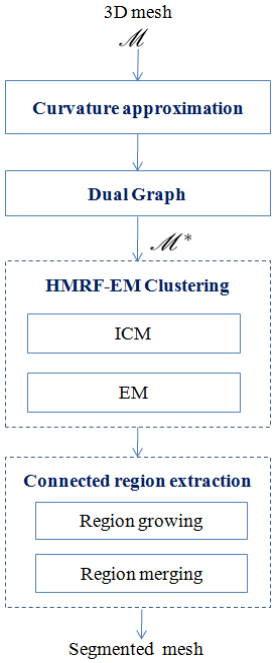


Figure 1: Method overview.

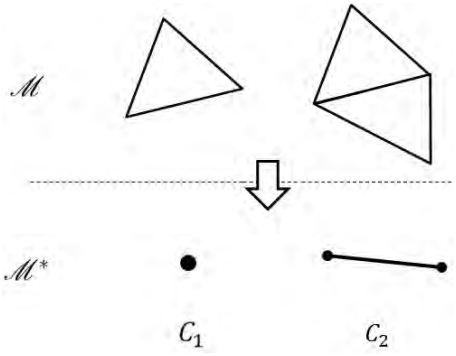


Figure 2: clique system.

3 THE HMRF-EM FRAMEWORK

3.1 Neighborhood and contextual relationship

In this work, we deal with the irrugular dual graph $\mathscr{M}^*(\mathrm{V}^*,\mathrm{E}^*)$ and we consider V^* as the set of sites denoted S. The MRF theory assumes that the sites are related to each other via a neighborhood system defined as $\mathscr{N}_s = \{t \in S, t \neq s \; et \; s \in \mathscr{N}_t\}$

In addition, a clique system is defined on S describing the configuration of all mutually neighboring sites or the site itself. In this work, we consider single-site clique and pair-site clique denoted respectively by C_1 and C_2 (see figure 2).

3.2 The Hidden Markov Random Field

Let $X = \{X_s; s \in S\}$ and $Y = \{Y_s; s \in S\}$ be two random fields corresponding respectively to labels and observations. The label field takes values in a discrete set L and the observation one in D.

In the segmentation context, we aim to estimate x a configuration of X based only on an observation y of Y. The underlying field X is non-observable, therefore the appropriate model is the Hidden Markov Random Field (HMRF).

A random field X is called a MRF on S with respect to the neighborhood system N if and only if $P(x) > 0$ and $P(x_s|x_{S-\{s\}}) = P(x_s|x_{\mathscr{N}_s})$, where $\mathscr{N}_s$ is the neighborhood of the site s. This last property expresses the behavior of the random variable on a site is determined by the neighboring random variables realisation and we can model practically all random variables whose mutual interdependence is resulting only from the combination of local interactions.

The Hammersley-Clifford's theorem [Bes74] establishes equivalence between MRF and Gibbs field. The distribution of X is given then by :

$$P(x) = \frac{1}{Z} exp(\frac{-U(x)}{T}) \qquad (1)$$

Where Z is the normalizing constant and $U(x)$ the energy function which is the sum of clique potentials $U_c(x)$ over all possible cliques C:

$$U(x) = \sum_{c \in C} U_c(x) \tag{2}$$

The energy U could be written as follows:

$$U(x_i = \ell | x_{\mathcal{N}_i}) = \sum_{j \in \mathcal{N}_i} \phi_{i,j} \delta(x_i, x_j) \tag{3}$$

Where $\delta(i,j) = \begin{cases} -1 & if \quad i = j \\ 1 & else \end{cases}$

and $\phi_{i,j} = \left\| e_{ij} \right\| \left| \alpha_{ij} \right|$

Where $\left\| e_{ij} \right\|$ is the length of the shared edge and $\left| \alpha_{ij} \right|$ is the absolute of the angle between the normals of the two faces sharing an edge (figure 3) :

$$\alpha_{i,j} = arccos \frac{n_{v_1 v_2 v_3} \cdot n_{v_2 v_4 v_3}}{\left\| n_{v_1 v_2 v_3} \right\| \left\| n_{v_2 v_4 v_3} \right\|} \tag{4}$$

Where $n_{v_1 v_2 v_3}$ et $n_{v_2 v_4 v_3}$ are the normals of the two faces and $\cdot$ is the scalar product of vectors. The normal $n_{v_1 v_2 v_3}$ is given by :

$$n_{v_1 v_2 v_3} = \frac{(v_2 - v_1) \times (v_3 - v_1)}{\left\| (v_2 - v_1) \times (v_3 - v_1) \right\|} \tag{5}$$

Where $\times$ is the vector product of two vectors.

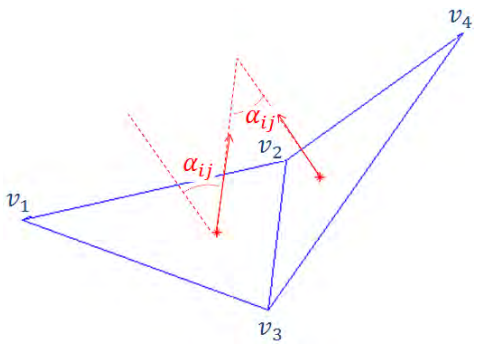


Figure 3: Dihedral angle: angle between the normals.

In a Bayesian context, we seek the solution of the maximum a posteriori expressing the most probable realization of the hidden variables given the observed one,

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)} \tag{6}$$

Assuming that $\{y_s\}_{s \in S}$ are conditionally independent

$$P(Y = y | X = x) = \prod_s P(Y_s = y_s | X_s = x_s) \tag{7}$$

The a posteriori probability became a function of the so called a posteriori energy

$$P(X = x | Y = y) \propto exp(LogP(Y|X) - U(x)) \propto exp(U(x|y)) \tag{8}$$

Where

$$U(x|y) = \sum_{s \in S} -LogP(y_s|x_s) + \sum_{c \in C} U_c(x) \tag{9}$$

And the maximum a posteriori estimator giving the labeling $\hat{x}$ is equivalent to

$$\hat{x}_{MAP} = argmin_x(U(x|y)) \tag{10}$$

The observation model is a multi-variate Gaussian one. In this work, we consider a 2 dimensional observation (maximum and minimun curvature). Thus the a posteriori energy is given by :

$$\begin{aligned} U(x|y) = \quad & \textstyle\sum_{s \in S} (-\frac{1}{2}(y_s - \mu_{x_s})^T \Sigma^{-1} (y_s - \mu_{x_s}) + Log(2\pi |\Sigma_{x_s}|^{1/2})) \\ & + \beta \textstyle\sum_{(s,t) \in C_2} \phi_{s,t} \delta(x_s, x_t) \end{aligned} \tag{11}$$

where μ_{x_s} and Σ_{x_s} are, respectively, the mean vector and the covariance matrix of class x_s.

To find the classification map $\hat{x}$ by the maximum a posteriori estimator corresponding to a minimization of the energy function $U(X|Y)$, a global optimization algorithm can be used such as simulated annealing. This algorithm was initiated by Kirkpatrick [Kir84] and adopted by Geman and Geman [Gem84] in the image processing context. The simulated annealing aims to find the global minimum of the energy that may have several local minimum. With analogy to thermodynamics, this algorithm incorporates a decreasing temperature parameter into the minimization procedure. For each temperature, it iteratively updates the energy function that will be accepted or rejected according to its probability which depends on the temperature parameter. This process is repeated until equilibrium state is reached. The simulated annealing algorithm ensures convergence to a global minimum energy but generates a large number of configurations as the temperature decreases which makes it a very time consuming algorithm. To overcome this disadvantage, we often use local algorithms such as the iterated conditional modes (ICM). This algorithm was proposed by Besag [Bes86], its principle is to iteratively update the sites labels based on the observation y and the current neighbors configuration of the each site. The new value $\hat{x}_s$ is obtained by maximizing the local probability $P(x_s | x_{\mathcal{N}_s}, y)$.

Since we are in a parametric context, an estimation of the Gaussian distribution parameters is required. In this work, we use an unsupervised algorithm of the maximum likelihood, the Expectation- Maximization (EM) algorithm.

3.3 The HMRF-EM algorithm

The combination of the ICM algorithm aiming to estimate the MAP resulting from the MRF theory and the EM algorithm gives an iterative algorithm called HMRF-EM and it can be resumed as follows:

In an iteration t:

- estimation of $\hat{x}$ by ICM
- estimation of $\theta_\ell(\mu_\ell, \Sigma_\ell)$

$$\hat{\mu}_\ell^{(t)} = \frac{\sum_{i \in S} P^{(t)}(\ell|y_i) y_i}{\sum_{i \in S} P^{(t)}(\ell|y_i)} \quad (12)$$

$$\hat{\Sigma}_\ell^{(t)} = \frac{\sum_{i \in S} P^{(t)}(\ell|y_i)(y_i - \hat{\mu}_\ell^{(t)})^T (y_i - \hat{\mu}_\ell^{(t)})}{\sum_{i \in S} P^{(t)}(\ell|y_i)} \quad (13)$$

where

$$P^{(t)}(\ell|y_i) = \frac{P^{(t)}(y_i|x_\ell, \theta_\ell) P^{(t)}(\ell|\hat{x}_{\mathcal{N}_i})}{P(y_i)} \quad (14)$$

The spatial information is encoded in the prior distribution $P^{(t)}(\ell|\hat{x}_{\mathcal{N}_i})$ given by :

$$P(\ell|\hat{x}_{N_i}) = \frac{exp(-U(\ell|V_i))}{\sum_{\xi \in L} exp(-U(\xi|V_i))} \quad (15)$$

4 CONNECTED REGION EXTRACTION

Once faces have been classified, a labeling operation is performed in order to extract connected significant regions. This procedure consists of a region growing step that produces a large set of connected regions which will be reduced with the following merging step (figure 4). In fact this latter one aims to merge similar neighbor regions according to a region distance measure [Lav05, Lav04]. In what follows, we briefly describe the region growing-region merging procedure.

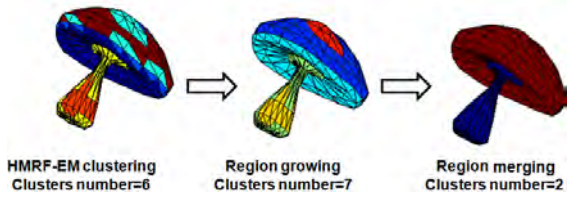


Figure 4: The region growing-merging process.

4.1 Region growing

Starting from seed triangles corresponding to faces which its neighbors belong to the same cluster, we iteratively expand the regions with new labels (each identified seed triangle is considered as a new region). Each triangle T_i that it is not yet labeled and has the same cluster as the seed triangle of the region that aggregate its neighbors joins this latter region. The growing step normally leads to holes between the identified connected regions (the triangles in the boundary of two regions are not labeled). In order to fulfill those holes, we assign a not labeled triangle to the most represented region in its neighbors and we repeat this process until every triangle is labeled. This step is called crack filling.

4.2 Region merging

The number of connected region produced by the growing step depends on the number of the clusters from the faces classification performed by the HMRF-EM algorithm. Generally, numerous small regions are identified and need to be merged with similar ones in order to have significant areas and for that a region adjacency graph (RAG) is used. The nodes on this graph represent the connected regions and the edges represent an adjacency between two regions. Edges are weighted with a similarity distance between the two corresponding regions. The region distance measure D_{ij} between two adjacent regions R_i and R_j is given by:

$$D_{i,j} = DC_{ij} \times N_{ij} \times S_{ij} \quad (16)$$

Where DC_{ij} is the curvature distance between R_i and R_j and equal to $\|C_i - C_{ij}\| + \|C_j - C_{ij}\|$, C_i and C_j are respectively the curvature values of R_i and R_j corresponding to the mean of the faces curvature of each region, and C_{ij} is the mean curvature of vertices on the boundary between R_i and R_j. The N_{ij} coefficient measures the nesting between the two corresponding regions which describes the spatial disposition of the regions and the S_{ij} coefficient allows to accelerate the merging of the smallest regions.
The processing of the graph reduction is as follows: at each iteration, the edge that has the smallest weight is eliminated and hence the corresponding regions are merged. Since the regions number is decreased by one, the graph is then updated and the process is repeated until the weight of the smallest edge is larger than a given threshold or a fixed number of regions is reached.

5 EXPERIMENTAL RESULTS

In figure 5, we compare our segmentation method with the Kmeans and EM clustering algorithm using 4 objects. In this experiment we set the number of clusters to 2 for the octopus and the dinosaur objects and to 4 for the vase and the eyeglass objects. Similar results are obtained for the octopus object identifying 9 regions (the head and the 8 arms). For the vase object the kmeans algorithm seems to provide finer decomposition than the EM and the HMRF-EM algorithm. In fact it allows to distinguish 6 regions rather than 3. We

can note that both partitions (3 or 6 region decomposition) correspond to meaningful parts of the object. Considering the eyeglass object, our method overperforms the kmeans and the EM algorithm since it enables to extract the two temples and the frame. In the case of the dinosaur object, the head was extracted only by the HMRF-EM method.

In order to show the efficiency of the proposed method for meshes that presents different curvature variations, we conducted experiments for 6 models (see figure 6). Our method provides good results for objects presenting low and medium curvature changes (first row in Figure 6). It generates non meaningful regions otherwise. In table 1, we measured the computational time for meshes presented in figure 6. The most time consuming step of the HMRF-EM algorithm is the classification step by the ICM algorithm where the prior energy is computed using neighboring triangles. Its complexity is equal to $O(\ell \times K \times N) = O(N)$, with ℓ is the upper bounds of the iterations number, K is the clusters number and N is the triangles number. Thus the complexity of the HMRF-EM algorithm is linearly dependent of the triangles number to be classified. As we can clearly notice in the table 1, for too dense meshes, the computation time is much higher than the one for simple meshes. The HMRF-EM algorithm was implemented with MATLAB. All experiments were performed on a PC with an Intel Core i7, CPU 2.5GHz and 8GB RAM.

3D Model	N	K	Processing time(s)
Mushroom	448	8	3.87
Octopus	2682	4	11.24
Bearing	7227	7	54.11
Fish	15142	4	71.31
Cup	30254	6	227.35
Bust	50456	10	645.25

Table 1: The computing time for different meshes

6 CONCLUSION

In this paper, a new 3D mesh segmentation based on the HMRF-EM framework has been proposed. The markov random field modelization is combined with the EM algorithm for parameters estimation. The definition of the prior model for this extended algorithm is based on dihedral angles which favorise the grouping of triangles having similar curvature values. After this clustering step, a region growing-merging process is applied in order to extract connected regions. Results show the efficiency of this extended framework. In a future work, we propose to use different type of attributes such as local mesh descriptors to improve segmentation results for more complex objects. In particular we aim to consider the 3D spectral information where the low frequencies correspond to the global shape while the hight frequencies contribute to the geometric details.

7 REFERENCES

[And07] Andersen, V. Smoothing 3D Meshes using Markov Random Fields. Master's thesis, IT University of Copenhagen, 2007.

[Bes74] Besag, J. Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society. Series B (Methodological), vol. 36, No. 2, pp. 192-236, 1974.

[Bes86] Besag, J. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, vol. 48, No. 3, pp. 259-302, 1986.

[Cha15] Changqing, Z., and Zhe, H., and Rynson, W., H., L., and Jianzhuang, L., and Hongbo, F. Sketch-based Shape Retrieval using Pyramid-of-Parts. CoRR abs/1502.04232,2015.

[Gem84] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI vol. 6, No. 6, pp. 721-741, 1984.

[Kal12] Kalogerakis, E., and Chaudhuri, S., and Koller, D., and Koltun, V. A Probabilistic Model for Component-based Shape Synthesis. ACM Trans. Graph., vol.31, No.4, pp.55:1-55:11, 2012.

[Kir84] Kirkpatrick, S. Optimization by simulated annealing : Quantitative studies. Journal of Statistical Physics, vol. 34, No. 5-6, pp. 975-986, 1984.

[Lav08] Lavoué, G., and Wolf, C. Markov random fields for improving 3D mesh analysis and segmentation. In Proceedings of the 1st Eurographics conference on 3D Object Retrieval (3DOR '08), Ioannis Pratikakis and Theoharis Theoharis (Eds.). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 25-32, 2008.

[Lav05] Lavoué, G., and Dupont, F., and Baskurt, A. A new CAD mesh segmentation method, based on curvature tensor analysis, Computer-Aided Design, Volume 37, Issue 10, pp. 975-987, 1 September 2005.

[Lav04] Lavoué, G., and Dupont, F., and Baskurt, A. Constant Curvature Region Decomposition of 3D-Mesh by a Mixed Approach Vertex-Triangle. WSCG, 2004.

[Mam09] Mamou, K., and Ghorbel, F. A simple and efficient approach for 3D mesh approximate convex decomposition. Image Processing (ICIP), 16th IEEE International Conference on. IEEE, pp. 3501-3504, 2009.

[Sha08] Shamir, A. A survey on mesh segmentation techniques. Computer Graphics Forum, vol.27, No.6, pp.1539-1556, 2008.

[Shl02] Shlafman, S., and Tal, A. and Katz, S. Metamorphosis of Polyhedral Surfaces using De-

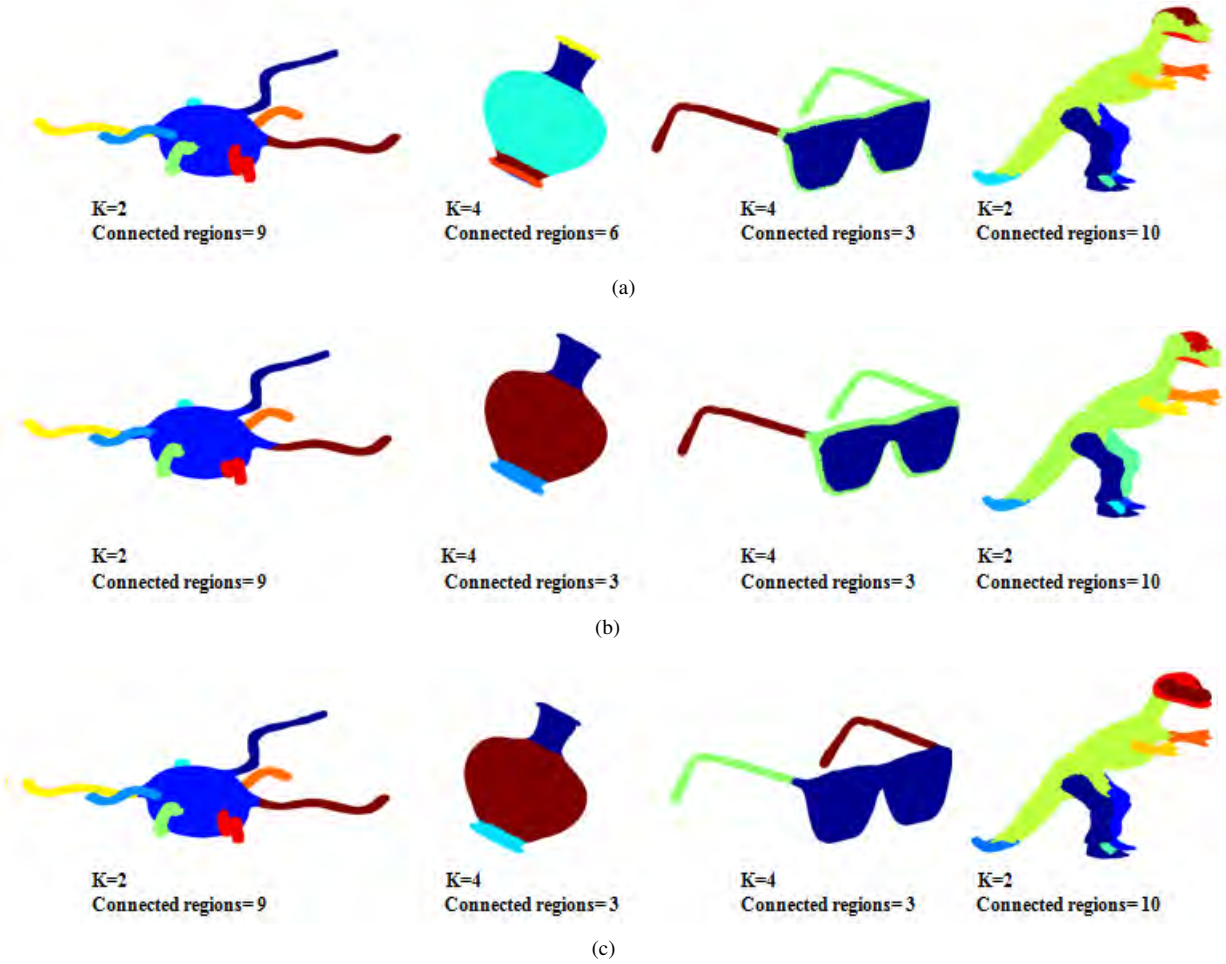


Figure 5: (a) K-Means and region growing, (b) the 2-dimensionnal EM and region growing, (c) the HMRF-EM and region growing.

composition. Computer Graphics Forum, vol.21, pp.219-228, 2002.

[Tsu14] Tsuchie, S., and Hosino, T., and Higashi, M. High-quality vertex clustering for surface mesh segmentation using Student- mixture model, Computer-Aided Design, Vol.46, pp. 69-78, 2014.

[Wil04] Willis, A., Speicher J., Cooper D. B. Surface sculpting with stochastic deformable 3D surfaces. In International Conference on Pattern Recognition, pp. 249-252, 2004.

[Zan01] Zang, Y., and Brady, M. , and Smith, S. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. IEEE Trans Med Imaging, vol. 20, No. 1, pp 45-57, 2001.

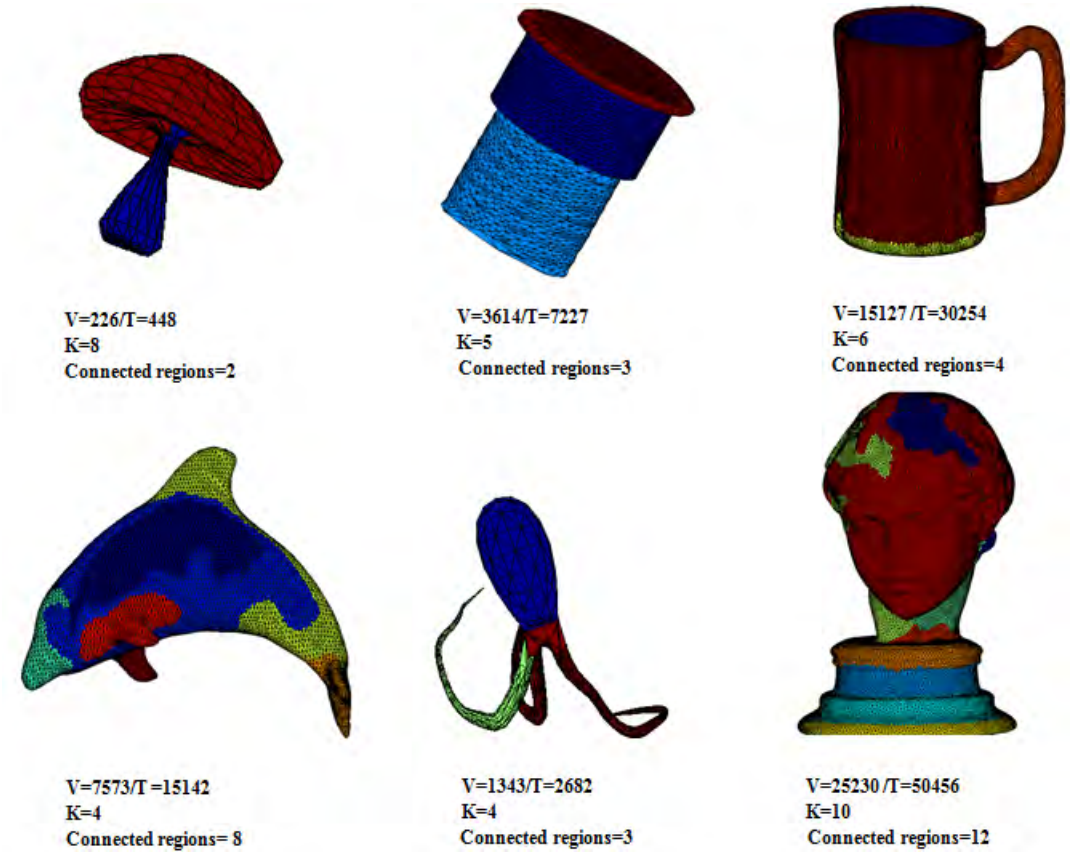


Figure 6: Segmentation results for objects with different curvature variations.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Fast and Robust Construction of 3D Architectural Models from 2D Plans

Jalaj Pandey
IIIT-Delhi, India
jalaj13043@iiitd.ac.in

Ojaswa Sharma
IIIT-Delhi, India
ojaswa@iiitd.ac.in

ABSTRACT

In this work we present a simple and robust method to create 3D building models from a set of architectural plans. Such plans are created for human readability and thus pose some problem in automatic creation of a 3D model. We suggest a semi-automated approach for plan cleaning and provide an algorithm for alignment and stacking of the plans followed by generation of 3D building model. We show results of our method on floor plans that generate complex 3D models in near real-time.

Keywords

Architectural plans, 3D model creation.

1 INTRODUCTION

3D architectural models find applications in a number of fields including virtual reality, scientific simulations, military training simulations to name a few. Efficient and accurate creation of 3D building models is therefore very important for a large scale model creation process. In this paper we propose a fast and robust method to combine architectural floor plans and generate a complete 3D building model.

Starting with a set of 2D architectural floor plans, manual construction of a 3D model is tedious and involves a multitude of steps including creation of individual building levels, constructing various floors, interior and exterior walls, and pillars. Various building levels need to be aligned so that they can then be stacked up to create the basic building structure and lastly operations such as slicing and bridging need to be performed to place separately created windows and doors. In this paper we reduce the manual labor and time required to create 3D building models by automating the process to a major extent. Our system allows the user to semi-automatically generate 3D building models from 2D plans resulting in significant reduction of the time required to create models.

2 BACKGROUND

The fundamental basis for generation of 3D models of a building are the 2D architectural floor plans. These plans contain enough information about the architectural design and geometry of the building. Usually the architectural plans exist in two formats: vector and raster (scan of hand drawn floor prints). For computer assisted 3D model creation of buildings, most existing approaches use vector form of the floor plans as in Lewis and Séquin [5] and Zhu et al. [13], while some use the scanned images of the floor plans, by first converting them into vector form using image processing algorithms (for example, Xuetao et al. [11] includes both types of systems).

Many systems make use of scanned floor plans/raster images and convert them into CAD files or vector images using various pattern recognition and image processing techniques in order to retrieve architectural information. An extensive survey by Xuetao et al. [11] provides a detailed description of converting raster images as performed by various systems. Approaches (like ours) that use a vector format, allow the system to recognise basic information about the plan since most of the plan geometry is grouped and labeled uniquely. Easier storage, data processing and layered structure of vector formats (like AutoCAD DWG/DFX) have made their use more popular in construction of architectural floor plans.

So et al. [10] proposed one of the first automated system for generation of 3D buildings. The system automated wall extrusion along with ceiling and floor construction. However, the constructed models require manual intervention in order to get usable results. The automation process was only able to reduce the time required to construct the final 3D building to some degree.

Moloo et al. [8] developed a software that uses a 3-phase recognition approach to generate 3D building from 2D floor plans. In their approach the authors rep-

resented a wall by grouping lines into bounding boxes. They aimed at automating the process of 3D building generation by analysing the 2D floor plan.

Or et al. [9] proposed a highly automated approach to generate 3D model from 2D floor plans. The system takes as input a set of raster images of floor plans and converts these into a set of vector images. Symbol recognition is used to identity symbols for doors and windows. Once Symbol recognition is completed, the system generates 3D buildings and imports these in Genesis 3D game engine. The system generates 3D buildings by analysing relationship of connected segments, that are created by parsing floor plans into connected segments.

Dosch et al. [3] presented a complete system that aims to reconstruct 3D buildings by analyzing scanned 2D architectural drawings. The system is divided into two steps, 2D modeling step in which they describe a robust graphics recognition algorithm that is used by the system for image processing and feature extraction. They describe this by dividing the raster 2D image into tiles, processing them individually and finally merging them after vectorization. For feature extraction the system makes use of a skeleton based approach, where extracted lines are represented as segments using polygon approximation technique. Next they propose a 3D modeling process that is used to match the reconstructed floors. Also the system also provides a user interface that is flexible and capable for human interaction.

Horna et al. [4] described a system that presents a four phase construction method. Their algorithm starts with removing geometrical inconsistencies by processing 2D edges. This is followed by topology generation using semantic information, and extrusion for 3D building construction. Lastly the floor is superimposed corresponding to upper and lower ceilings that are linked by stairways to construct the final model. The complete topological model constructed by the system expresses incidence and adjacency relations between elements. Further, the system also associates semantic information with all volumes for specifying structures like rooms and walls. The system takes the semantic information into account during extrusion to 3D.

Ahmed et al. [1] proposed a complete system for automated floor plan analysis. The system helps to apply and improve the present processing methods, further it also introduces preprocessing methods that improve the performance of the system. Some of the techniques used is the differentiation between thick, medium and thin lines and removing components that lie outside the convex hull of the outer walls.

The Berkeley WALKTHRU system , developed by Lewis and Séquin [5], is a semi-automatic system capable of generating 3D polyhedral buildings from AutoCAD DXF format with minimal user interaction. This system uses the concept of portals and spaces for stacking of floors in a multi-storey building. The models generated are a solid representation of the real buildings that can be used to develop a virtual walkthroughs and computer rendering. According to Zhu et al. [13], even though this approach involves minimal human intervention, it takes days to create a complex 3D model.

Zhu et al. [13] proposed a system to construct 3D building models automatically from vector floor plans by analysing semantic information and geometry from the plans. Their system made use of defined axes in each floor plan for stacking. The basic idea was to align every floor to the first floor plan by matching/equating their respective axes. Two axes are equal when they are of the same type and have the same label. The complexity of this algorithm is $O(n^2)$. The authors describe several interesting 3D building creation methods. The one by Zhi et al. [12] automatically creates a fire evacuation building simulator model, while the one by Domínguez et al. [2] introduces an interesting semi-automatic approach that detects the topology of building floors. Lastly the ones by Lu et al. [6, 7] indicate component types by making use of architectural drawings without labels and computer drawn construction structural drawings.

In this paper we present a system that allows the user to create 3D architectural models. Our novel contribution to the state-of-the-art is a robust algorithm for alignment and stacking of floor plans in absence of a common 2D coordinate system across plans. We provide a semi-automated end-to-end approach for creating usable 3D building models.

3 APPROACH

Our 3D model construction approach takes 2D architectural plans in a CAD vector format. The algorithm performs plan cleaning based on object attributes. The cleaned floor plans are then aligned and stacked by the system at their respective heights in the building. The building level heights are input parameters to our system. This is followed by generation of 3D model of the building by extruding the floor horizontally and the walls and other entities in the building vertically. Finally the process ends by performing a slicing operation that is used for placement of doors and windows. Algorithm 1 summarises our four-step approach to 3D building creation.

Next we describe steps of our proposed approach. The system is supplied with raw architecture vector plans. The process for generation of the 3D model in our system is divided into four basic steps.

3.1 Plan cleaning

The first step of the approach is to clean the raw vector plans so that they can be supplied to the system for fur-

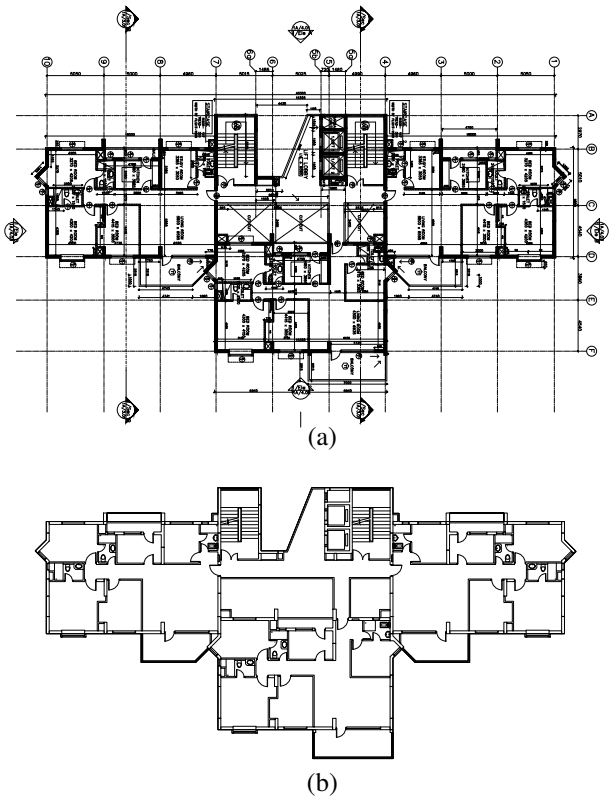
Input: 2D floor plans, building parameters
Output: 3D building model
PLAN CLEANING
Removal of text, and annotation tags
Detailing identification and removal
PLAN ALIGNMENT AND STACKING
Creation of tree of architectural elements
Alignment using elements of structural stability
FLOOR GENERATION AND EXTRUSION
Floor generation by horiozontal filling
Vertical extrusion of walls
SLICING AND WINDOW PLACEMENT
Creation of space for windows and doors

Algorithm 1: 3D building generation approach.

ther processing. An architectural plan consists of multiple elements required in a building design. Apart from these, the plans also contain text and supportive layout elements (like icons, grid lines, and guides). Only a subset of these elements is required for 3D model creation. Such elements include exterior and interior walls, columns, and elevator spaces. Plan cleaning is a complicated task and an automated approach requires machine recognition of these elements (see [1, 2, 3]). As a result, a complete analysis and recognition is not possible in all scenarios, and some geometrical elements needs user intervention to be removed in the cleaning process. We adopt a semi-automated approach to cleaning plans by combining geometric analysis with minimal user interaction. Earlier works have also resorted to processing raster 2D plans, which in our opinion is not only difficult but also leads to inaccuracies in final 3D reconstruction. We operate on vector 2D plans and assume that various elements have been tagged in some way by the architect who authored the plans.

Our plan cleaning follows a two step procedure. In the first step, we identify relevant elements within a plan by analysing the tags attached to each one of these and grouping similar elements together. Similarity of elements in this context refers to elements/objects falling under the same category such as doors and window, lifts (defined as portals in [2, 5]), and walls. Such an analysis is carried out by means of string matching with regular expressions on element text attributes. The system tries to extract element type attributes automatically with this analysis and presents it to the user for verification. The user can quickly correct attribute names or reject erroneous results. The system then removes unnecessary attributes and renames required attributes consistently.

Once the attributes are processed and elements are grouped, the system performs a geometric analysis to clean individual elements. This step is similar to the processing proposed in [2, 5], and primarily looks at geometric integrity of various elements required for extrusion in a subsequent step. Figure 1 shows result of cleaning on one of the plans.



(a)

(b)

Figure 1: Result of cleaning on a highly detailed floor plan.

3.2 Plan alignment and stacking

Aligning plans of multi-floored models poses a challenge in generation of 3D building models [5, 13]. The process is complicated by the fact that floor plans are designed separately for multi floor buildings, thus each floor plan is designed with its own local 2D coordinate system. In our experience, these local coordinate systems do not always align with each other and thus it is not straightforward to stack up these plans for extrusion. Our system resolves the alignment problem by creating a hierarchical parent-child relationship between elements of the floor plan, which is the pivotal part of the algorithm.

The system prompts the user to identify architectural elements that identify structural stability and use those to align all floor plans. Such elements will always be aligned in any given building to satisfy structural stability (these include load bearing pillars, lift spaces and stairs). Such elements are grouped together as root element of the hierarchy in a plan and other elements are arranged below these. In our tree hierarchy, the stability element that is identified, acts as a pivot to which all other layers are attached. Consider this as a tree with

one root and the remaining layers of the floor plan as its child nodes. The only property this parent-child relationship follows is that whenever the parent moves the child nodes move relative to it, thus not changing their respective positions among themselves, but if the user moves any child node, the parent or other nodes remain unaffected. The system iteratively aligns all floor plans by estimating a rigid transformation (including translation and rotation) between root elements from two plans at a time. All elements within a plan are transformed to a common coordinate system where all plans are aligned to each other. This gives us a robust approach to align plans under any circumstance. Many-a-times the floor plans are complex and asymmetric, but with our approach those are handled very well.

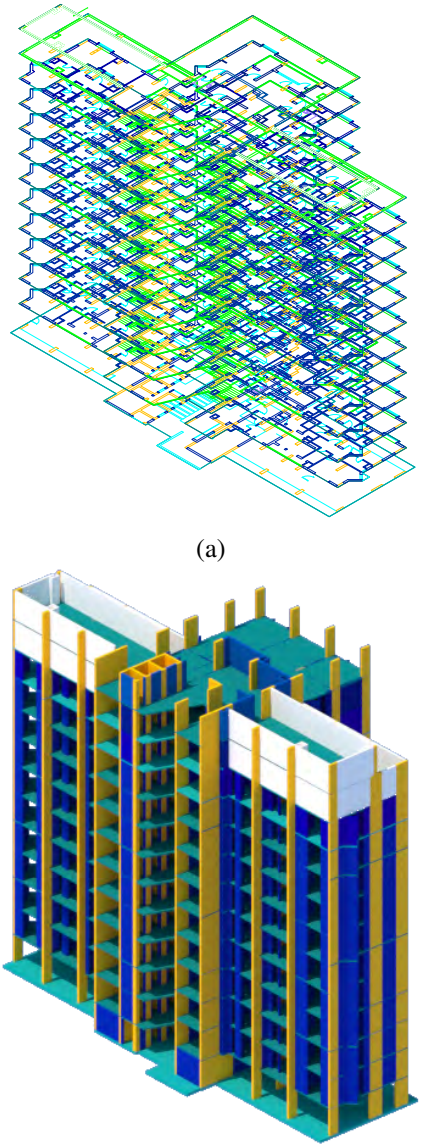
The floor plans are stacked once they are aligned. In our system we again use the tree root to stack the floor plans. The system is supplied the respective heights of all the floor plans from which each plan is transformed and placed at its respective height.

3.3 Floor generation and extrusion

We assume that geometric elements in a 3D building model are composed of extruded elements. Extrusion is a process of creating a cylindrical or planar element from its 2D footprint (a circle or a line segment) by extending it in the third dimension. For creation of a 3D model from stacked 2D floor plans, two types of geometries are required: horizontal floors and vertical walls. We utilise region fill and extrusion for creation of these.

Our system first generates floor polygons from wall boundaries by creating planar horizontal faces. Exterior and interior walls (and load bearing pillars) are then extruded vertically. This is performed completely automatically and creates a 3D model with solid faces. The system is capable of creating planar and cylindrical geometry by extrusion. Walls, floors and ceilings generated by our system have finite width (and are not merely thin planar geometry). In special cases, it can also handle surfaces of revolution by minimal modification (e.g. spherical geometry for tombs). Roof of a building is generated by replicating the plan of the top floor and placing it at the required height. A simple hipped roof, as seen in contemporary European-style buildings, may be generated by creating slanted planes guided by the top floor elevation. Figure 2 shows a complete extruded 3D model from stacked floor plans (the roof is removed to improve visualization).

Complicated elements like staircases require special treatment. These can be generated by combining information from both the plan and the elevation of a floor. A floor plan usually depicts steps in a staircase, while the elevation contains information about the slope and stride. Staircase generation is not currently implemented in our system.



(a)

(b)

Figure 2: Creation of extruded 3D building model. (a) aligned and Stacked 2D floor plans in 3D space, (b) extruded 3D building model.

3.4 Slicing and window placement

The 3D model constructed so far comprises of solid walls with no windows and doors. For a complete building creation, these structural elements are important. Our system handles window and door creation by slicing the 3D extruded model horizontally at various levels and creating placeholders for windows and doors. Input parameter to slicing is heights of door and window elements. The system extracts the width from the respective floor plan and creates a placeholder geometry for the window/door element. Detailed window and door 3D models can then be easily added to these models by the user.

4 RESULTS

We show results of our system with a set of complex floor plans of various campus buildings. We implemented our system completely in Autodesk 3ds Max using the MAXScript programming interface. All of our result are produced on an Intel Core i7 2.4 GHz processor with 6 GB memory (on a laptop computer).

In the results shown below, we purposefully omitted the building roofs for better visualisation of results and to highlight complexity of our models. Figure 3 shows two views of the 3D model of Academic block that consists of three distinct wings arranged at an angle to each other. Other reconstructions include a horseshoe shaped hostel building (see Figure 4) and the student center (see Figure 5). Figure 2(b) illustrates reconstruction of a 11-storey residence building.

Table 1 shows runtime information in seconds for generation of full 3D model using available floor plans. We note that these numbers depend on the complexity and number of the floor plans. These numbers show that our system is capable of generating highly detailed 3D models in near real-time on commodity mobile computer hardware.

Various tasks in 3D model creation can be automated to various degrees. We achieve a high automation in several intermediate steps. This is summarised in Table 2 along with comparison across existing methods.

5 CONCLUSION

In this paper, we presented a simple and robust system for 3D model generation from a set of unaligned 2D architectural floor plans. We illustrated our alignment algorithm that is a pivotal part of our system and allows us to align the floor plans and thus proceed with more general operations like extrusion to construct the 3D models. Currently our algorithm cannot identify the stability element on its own, which we would like to address in future. Also we plan to provide texture support in next version of our system. Generation of non-vertical complex structures is challenging. Such architectural geometries need to be handled on a case-by-case basis. In general, these may be split into multiple parts which may either be extruded along a non-standard axis or modelled by minimal surfaces. We would like to address these challenges in future.

6 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

7 REFERENCES

[1] Sheraz Ahmed, Marcus Liwicki, Markus Weber, and Andreas Dengel. Improved automatic analysis of architectural floor plans. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 864–869. IEEE, 2011.

[2] B Domínguez, ÁL García, and Francisco R Feito. Semiautomatic detection of floor topology from CAD architectural drawings. *Computer-Aided Design*, 44(5):367–378, 2012.

[3] Philippe Dosch, Karl Tombre, Christian Ah-Soon, and Gérald Masini. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3(2):102–116, 2000.

[4] Sebastien Horna, Guillaume Damiand, Daniel Meneveaux, and Yves Bertrand. Building 3D indoor scenes topology from 2D architectural plans. In *GRAPP (GM/R)*, pages 37–44. Citeseer, 2007.

[5] Rick Lewis and Carlo Séquin. Generation of 3D building models from 2D architectural plans. *Computer-Aided Design*, 30(10):765–779, 1998.

[6] Tong Lu, Chiew-Lan Tai, Feng Su, and Shijie Cai. A new recognition model for electronic architectural drawings. *Computer-Aided Design*, 37(10):1053–1069, 2005.

[7] Tong Lu, Huafei Yang, Ruoyu Yang, and Shijie Cai. Automatic analysis and integration of architectural drawings. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(1):31–47, 2007.

[8] Raj Kishen Moloo, Muhammad Ajmal Sheik Dawood, and Abu Salmaan Auleear. 3-phase recognition approach to pseudo 3D building generation from 2D floor plan. *arXiv preprint arXiv:1107.3680*, 2011.

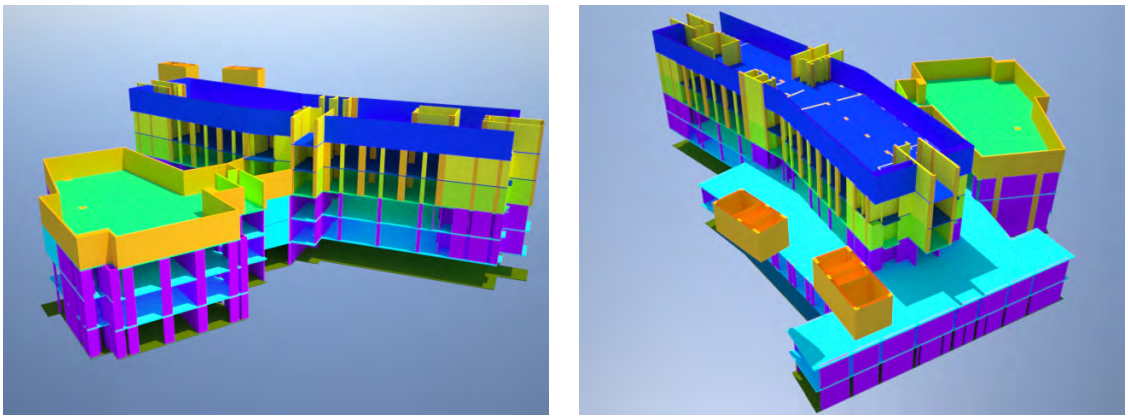
[9] Siu-Hang Or, Kin-Hong Wong, Ying-kin Yu, Michael Mingyuan Chang, and H Kong. Highly automatic approach to architectural floorplan image understanding & model generation. *Pattern Recognition*, pages 25–32, 2005.

[10] Clifford So, George Baciu, and Hanqiu Sun. Reconstruction of 3D virtual buildings from 2D architectural floor plans. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 17–23. ACM, 1998.

[11] Xuetao Yin, Peter Wonka, and Anshuman Razdan. Generating 3D building models from architectural drawings: A survey. *IEEE Computer Graphics and Applications*, (1):20–30, 2009.

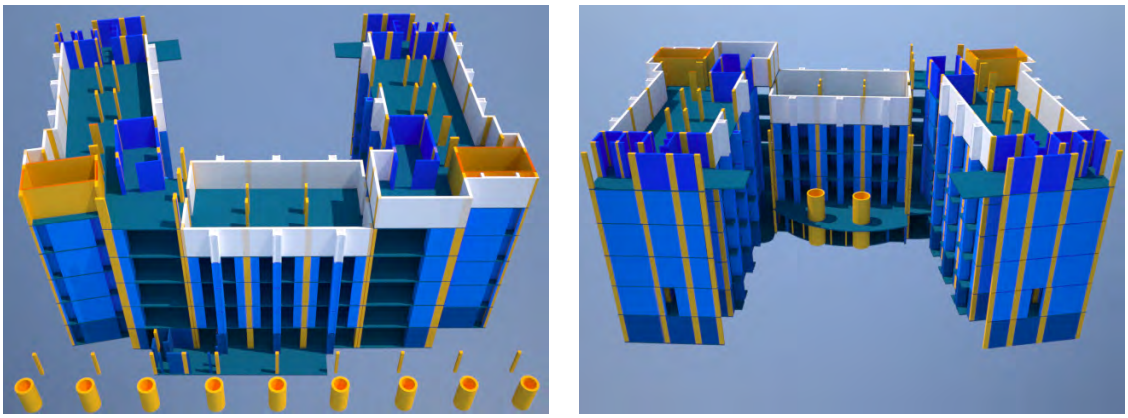
[12] GS Zhi, SM Lo, and Z Fang. A graph-based algorithm for extracting units and loops from architectural floor plans for a building evacuation model. *Computer-Aided Design*, 35(1):1–14, 2003.

[13] Junfang Zhu, Hui Zhang, and Yamei Wen. A new reconstruction method for 3D buildings from 2D



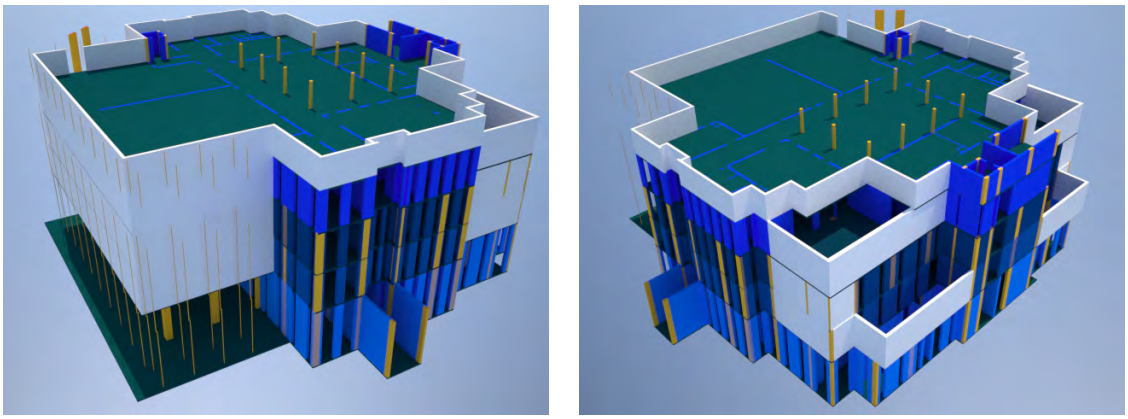
(a) (b)

Figure 3: Reconstructed Academic block model (a) front view, (b) back view.



(a) (b)

Figure 4: Reconstructed Hostel building (a) front view, (b) back view.



(a) (b)

Figure 5: Reconstructed Student center (a) front view, (b) back view.

Building	*# Input plans*	*# Output triangles*	*Without slicing (sec.)*	*With slicing (sec.)*
Faculty residence	13	130,498	15.280	29.160
Student center	5	41,166	7.083	10.183
Hostel	6	110,312	6.810	13.342
Academic block	6	86,984	7.302	14.784

Table 1: Construction times for entire model generation.

Process	*Lewis and Séquin* [5]	*So et al.* [10]	*Lu et al.* [7]	*Dosch et al.* [3]	*Or et al.* [9]	*Ours*
Vector plan input	Yes	Yes	Yes	No	No	Yes
Plan cleaning	Semi-Automatic	No	Manual	Manual	No	Semi-automatic
Plan alignment and stacking	Semi-automatic	No	No	Automatic	No	Semi-automatic
Floor generation and extrusion	Automatic	Semi-automatic	Automatic	Semi-automatic	Automatic	Automatic
Slicing and window placement	Automatic	Manual	Automatic	Automatic	Automatic	Automatic
Overall automation	High	Low	Medium	High	Medium	High

Table 2: Comparison of degree of automation with various approaches.

vector floor plan. *Computer-Aided Design and Applications*, 11(6):704–714, 2014.

Automatic segmentation of cervical cells in Pap smear images

Omelkhir Boughzala
Laboratory of Technology and Medical imaging (LTIM)
Monastir 5000, Tunisia
University of Monastir
omelkhirboughzala @ gmail.com

Lamia Guesmi
Laboratory of Technology and Medical imaging (LTIM)
Monastir 5000, Tunisia
University of Monastir
lamia_guesmi0107@ yahoo.com

Asma Ben Abdallah
Laboratory of Technology and Medical imaging (LTIM)
Monastir 5000, Tunisia
University of Monastir
assoumaba@ yahoo.com

Mohamed Hédi Bedoui
Laboratory of Technology and Medical imaging (LTIM)
Monastir 5000, Tunisia
University of Monastir
MedHedi.Bedoui@ fmm.rnu.tn

ABSTRACT

In the context of medical diagnosis by image analysis, segmentation is the most critical step in image processing. The problem of image segmentation has been studied for years and many methods have been suggested in the literature. However, there is not yet any automatic method able to correctly process any type of image. In this work, we present an automated method for cell segmentation in Pap smear images. The automatic analysis of Pap smear images is one of the most interesting fields in medical image processing. The object of this paper is to present the strategy of the first part of the system segmentation. It is based on a segmentation of color images tested with different classical color spaces, namely RGB, L*a*b, HSV, and YCbCr, to select the best color space using k-means clustering to separate groups of objects. The k means clustering treats each object as having a location in space. The method is aimed at developing an automated Pap smear analysis system which can help cytotechnologists reduce examination time in pap screening process.

Keywords

Pap smear, medical image, processing, cervical cancer detection, cytologic screening, K-means clustring

1. INTRODUCTION

The Pap smear is a technique of cervical screening used to detect pre-cancerous changes of the uterine cervix. It is not designed to find cancer or any abnormalities of any organs. These changes are called cervical intraepithelial neoplasia (CIN) [O14]. They are obtained by opening the vaginal canal with a speculum and scraping the cervix with a wooden stick and a tiny brush, which are scratched on a glass slide in order to collect the cells. The collected cells are then examined by a cytotechnologist.

At present slide examination of cervical cytology image is performed manually. The cyto-technologist looks at a glass smear under the microscope and analyses the full image in order to determine the presence of disease. He / She is involved in the diagnosis of cancer, pre-cancerous lesion, benign tumors and infections from various body sites. Cytotechnologists work under the direction of a physician called a pathologist, a medical doctor who specializes in the study of diseases and determines the nature and cause of the problem.

Most patients never meet the pathologist or the cytotechnologist who evaluates their samples, yet the treatment of their illness may depend on the work of the

pathologist- cytotechnologist team. Screening consists in locating and visually assessing all the present cells on a slide. This mainly aimed at detecting abnormal or suspicious cells to reach a diagnosis. This is of a major interest for the cytotechnologist whose diagnosis will depend on the good recognition of the abnormal or suspicious cells during screening. This cytological manual screening is described as intense and complex. The results are based on the point of view of a human being. It is often tiring to screen for relatively small and abnormal cells, which is time-consuming and requires high concentration. [B03].For all these factors, some errors appeared and can cause false negatives [NAH97]. These errors are considered as

being inseparable of the process of manual screening. A promising approach is to help the cyto-pathologist in his / her search for abnormal cells on a glass. An automatic system could contribute to the detection of screening errors and thus allow a better reliability of the diagnosis. Such a system is programmed to verify the results of conventional screening and possible false negatives. This would prevent delays allow considerable time gain.

The automatic segmentation system of cervical cells in Pap smear images is the most crucial step for any system. Segmentation is aimed at detecting the cells along with their nuclei and cytoplasms (see figure1) [LEC+98]. The shape and report area of the cytoplasm and nucleus are two important factors in detecting pre-cancerous changes in the uterine cervix [CHL+13]. Due to the complexities of cells many studies have focused on segmentation of cytopathological images [PN12] [DUK08].In the literature, some methods used thresholding techniques [KSW07] [LN07]. Many methods of segmentation of the nucleus and the cytoplasm focused on edge detection [LCC09], CHL+13]. For example [RNRT12][PN11] adapted the mathematical morphology to segment both the nucleus and the cytoplasm of a single cell. Other works utilized the genetic algorithm[LH03] and the deformable template [GP00]. A watershed transform has been applied in [MKI]. Recently Sajeena T A and Jereesh A S proposed a framework for automatic analysis of single cellular pap smear slides[SJ15].Fuzzy C-means (FCM) clustering technique is proposed for single cell segmentation[KSN15].

In this paper, we propose a method of 2D color cytological image segmentation using the color information as a priori information. The full image is segmented into 3 regions: the nuclei, the cytoplasm and the background. When we consider color image segmentation, choosing a proper color space becomes the most important issue. In this work, a segmentation of color images is tested with different classical color spaces: RGB, HSV, L*a*b, and YCbCr, to select the best color space for the considered kind of images. The segmentation process is based on the K-means segmentation technique

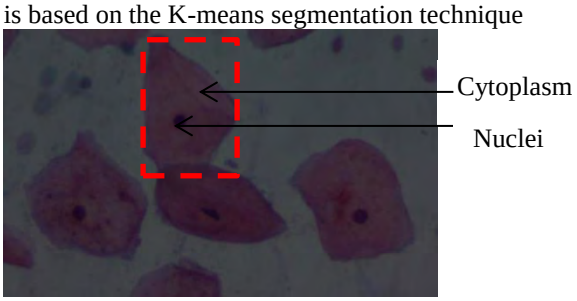


Figure 1 Macroscopic image of cytology

2. Method

The slides are examined by a microscope to which a color camera is fixed. The obtained images are color images of 1030X1300 pixels. It is necessary for us to isolate their cytoplasm and nuclei at the same time: the cytoplasm to obtain information to characterize isolated cells and the nuclei to characterize the cell and estimate the wickedness. The cytoplasms and the nuclei will help the recognition of different cells. To carry out the segmentation step we should know the nature and the context of the images. Our color images present cells from the cytology of the cervical sample colored by the international coloration standard of Papanicolaou[GK96][MCL91]. Cells have a blue nuclei and a green cytoplasm with the exception of the red blood cells which are totally colored in red. The spatial configuration of the cells and their color are extremely variable. There can be isolated, attached but also heap cells which can overlap (nuclei or cytoplasm). The color of the nuclei can vary from very pale blue to very dark blue. This big variety in the spatial configuration and the color of cells raises problems of segmentation and requires a method of fine and strong segmentation at the same time

2.1 General presentation of the method

The strategy of segmentation which we organized is the segmentation of color images by the k-means method.

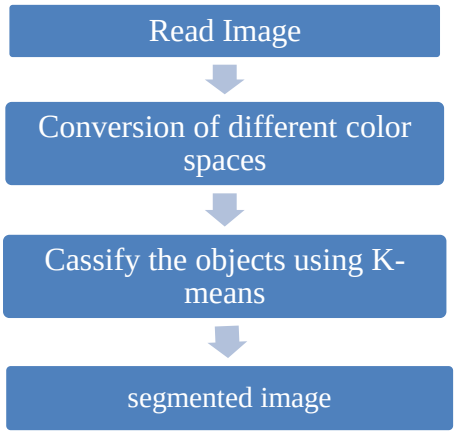


Figure 2 the color image segmentation strategy Pap smear cytology

2.2 Conversion in different spaces of colors

A color space combines numbers to the visible colors. Generally, it represents a color by a triplet of values. The visible colors can be seen as belonging to a three-dimensional space. There are many color spaces, each with its properties. The conversion of the image in different spaces of colors is the preliminary step to the achievement of the segmentation. Each color space has interesting properties and presents an interest for such an application. Generally the choice of the color space is done when the segmentation method is established. Several works were focused on comparing different color spaces [KEHT14] [LEC+98][LEC03]. Others chose to use multiple color spaces at the same time [MK12] [BA12] [M08].

2.1.1 RGB color space

Many color spaces are in use today. For pictures captured by digital cameras, the most popular one is the RGB model. This is an additive color model in which the colors red, green, and blue are combined together in different manners to reproduce a broad array of colors. This color space-based segmentation is not accurate for computer vision applications. The RGB color is device-dependent i.e the same signal or image can be viewed differently with different devices, Nunobiki and all reported the usefulness of RGB color specification in analyzing the variation of color properties for Papnicolaou-stained cervical smears. [NST+02].

2.1.2 HSV color space

The HSV color space (Hue, saturation, Value) define a model in terms of its components. The space has the ability to separate the intensity of the color information hue and saturation. For this reason it has been adopted for processing images having brightness variation characteristics. Many works use this space [OTDC02]. The conversion from RGB to HSV

$$H=\begin{cases} 0 & \text{if } \max(R,G,B) \\ 60^{*}\frac{G-B}{\max(R,G,B)-\min(R,G,B)} & \text{if } R=\max(R,G,B) \text{ etG}\geq B \\ 60^{*}\frac{G-B}{\max(R,G,B)-\min(R,G,B)}+360 & \text{ifR}=\max(R,G,B) \text{ etG}<B \\ 60^{*}\frac{R-G}{\max(R,G,B)-\min(R,G,B)}+240 & \text{ifB}=\max(R,G,B) \\ 60^{*}\frac{B-R}{\max(R,G,B)-\min(R,G,B)}+120 & \text{ifG}=\max(R,G,B) \end{cases} \quad (1)$$

$$S=\begin{cases} \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} & \text{if } \max(R,G,B)\neq 0 \\ 0 & \text{if } \max(R,G,B)=0 \end{cases}$$

$$V=\max(R,G,B)$$

2.1.3 L*a*b color space

A L*a*b color space is a color opponent space with dimension L for lightness, the a* layer indicates where the color falls along the red green axis, and b* layer indicates where the color falls along the blue-yellow axis. The L*a*b* color space includes all perceivable colors, which means that its gamut exceeds those of the RGB and CMYK color models (for example, ProPhoto RGB includes about 90% all perceivable colors). The most important feature of this color space is that it ,is device-independent, that is to say provides us with the opportunity to communicate different colors across different devices [RKV12]. The solution to convert images from the RGB space to the L*a*b* color space is given by the following formula. The conversion from RGB to XYZ is:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}=\begin{bmatrix} 0.618 & 0.177 & 0.205 \\ 0.299 & 0.587 & 0.114 \\ 0 & 0.056 & 0.944 \end{bmatrix}*\begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2)$$

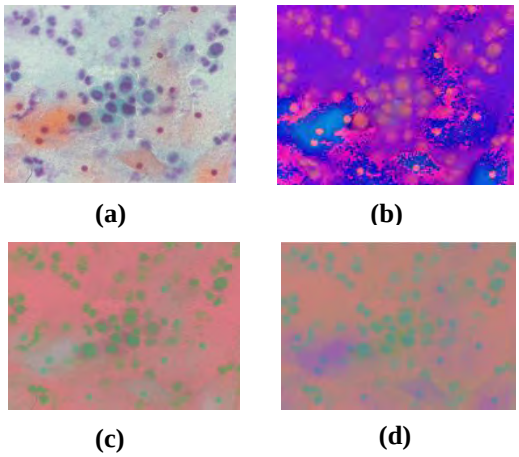
The transformation from XYZ to Lab is performed with the following equations:

$$\begin{aligned} L^{*} &= 116(Y/Y_n)^{\frac{1}{3}}-16 \\ a^{*} &= 500\left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}}-\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right] \quad (3) \\ b^{*} &= 200\left[(Y/Y_n)^{\frac{1}{3}}-(Z/Z_n)^{\frac{1}{3}}\right] \end{aligned}$$

2.1.4 YCbCr color space

The YCbCr color space is used in digital video image processing. It represents colors in terms of one luminance component (Y) and two chrominance components (Cb and Cr). The Cb component is the difference between the blue component and a value, where Cr is the chrominance red component. In contrast to RGB, the YCbCr color space is luminance- independent, that is why it gives better performance.[BTP+11]The transformation used to convert from RGB to YCbCr color space is shown in equation (4):

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}=\begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}*\begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix}*\begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$



(a) **(b)** **(c)** **(d)**

Figure 3 (a) Image RGB (b) Image HSV (c) Image L*a*b (d) Image YCbCr

2.3 Image Segmentation Using K-means

Clustering is a process to separate groups of objects. The algorithm k-means is the algorithm of the most known and most used clustering, because of its simplicity of implementation [MK12]. K-means clustering treats each object as having a location in space. It allows to partition the data of an image K clusters. Contrary to other hierarchical methods, which create a structure in “tree of clusters “to describe the groupings, the k-means creates only a single level of clusters. The algorithm sends back a partition of the data, in which similar objects are placed as close as possible to each other in the same cluster, and the different ones are placed as far as possible in another cluster. The k-means is an iterative algorithm which minimizes the sum of the distances between every object and the centroid of its

cluster. The final result depends on the centroids' initial position. Therefore, the centroids must be placed as far as possible from each other so as to optimize the algorithm. K-means changes the objects of cluster until the sum cannot decrease anymore. The result is a set of clusters that are compact and clearly separated provided that the best K value of the number of clusters is chosen. The main stages of the algorithm k-means are (see figure 4) :

1. Random choice of the initial position of K clusters.
2. Allocate objects to a cluster following a criterion of minimization of the distances (generally according to a measure of Euclidian distance).
3. Once all the objects are placed, recalculate K - centroids.
4. Repeat stages 2 and 3 until no more reallocations are made.

We are going to look at each of the color spaces according to their influence on the algorithm of k-average and then we will present a method for choosing a color space.

Since the color information exists in the different color space our objects are pixels with channels values. Use k-means to place the objects into three clusters using the Euclidean distance metric. For every object in the input, k-means returns an index corresponding to a cluster. Label every pixel in the image with its cluster index figure 5.

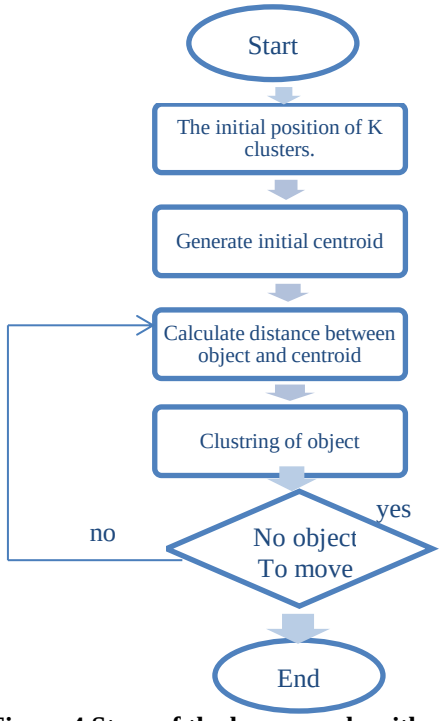


Figure 4 Steps of the k-means algorithm

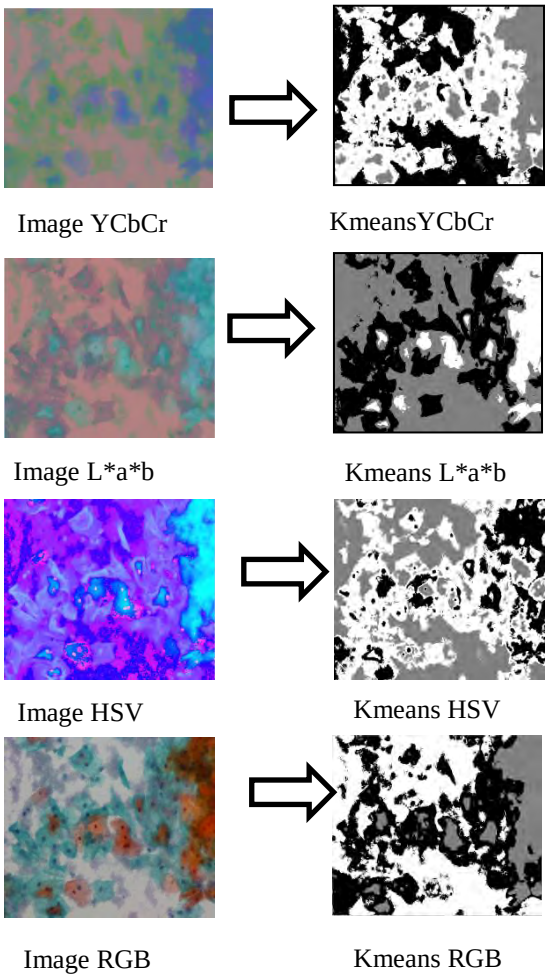


Figure 4 Example of Kmeans segmentation with different color space YCbCr,L*a*b, HSV and RGB.

3. Results and discussion

The automatic K-means technique was experimentally tested using a dataset of 9 different images. It was applied with different color space models, including RGB, HSV, L*a*b and YCbCr. The features that identify each image pixel are only the values of its three components in the selected color space. We took the k values (number of clusters) as 3 for the K-means algorithm and the distance metric chosen is cosine. For every object in input, k means returns an index corresponding to a cluster. Label every pixel in the image with its cluster index. We can separate objects in an image by color, which will result in three images (color background, color cytoplasm and color nuclei) see figure 6.

For evaluation, a mathematic expression -described in (4) and (5) is proposed to obtain the percentage of amount of cells that are in the input image. The percentage nuclei and cytoplasm segmentation are calculated as the fraction of pixels with segmentations of K-means technique divided by the total number of pixels in ground truth. The results obtained are shown in tables 1 and 2. The color space

RGB arises as giving the highest one (see Table 1). This color space is therefore more suitable for the segmentation of the nuclei. The color space RGB is done on representative images of the nuclei of the cells. The nuclei of the cells have an extremely variable color. We can see also that the RGB space is the one that obtains better results for the segmentation of the cytoplasm table 2

$$\text{nuclei } \% = \frac{nucleicolorpixels}{imagepixelsingroundtruth} * 100 \quad (4)$$

$$\text{cytoplasm } \% = \frac{\text{cytoplasm color pixels}}{\text{image pixels in ground truth}} * 100 \quad (5)$$

Table 1: Experimental nuclei segmentation results on different color spaces using K-means.

Image	cell percentage %			
	RGB	HSV	L*a*b	YCbCr
1	98.43	34.43	99.06	53.94
2	40.63	36.27	36.21	53.79
3	79.70	68.31	65.25	97.20
4	14.72	14.50	14.38	12.17
5	78.92	61.10	79.80	26.63
6	52.59	45.39	56.38	98.93
7	93.39	47.51	52.56	70,59
8	43.67	99.40	40.04	43.51
9	80.16	37.40	53.64	85.93
Avg	64.69	49.36	55.25	60.29

Table 2 Experimental cytoplasm segmentation results on different color spaces using K-means

Image	cell percentage %			
	RGB	HSV	L*a*b	YCbCr
1	99.06	40.74	94.83	96.64
2	61.71	65.21	64.62	99.79
3	99.02	98.30	99.23	86.32
4	21.52	25.57	20.65	12.82
5	99.04	98.2	97.60	98.20
6	68.17	93.91	62.79	45.34
7	98.01	92.72	99.04	61.27
8	97.14	50.00	82.77	77.65
9	92.45	87.40	97.03	89.51
Avg	81.79	72.45	79.84	74.17

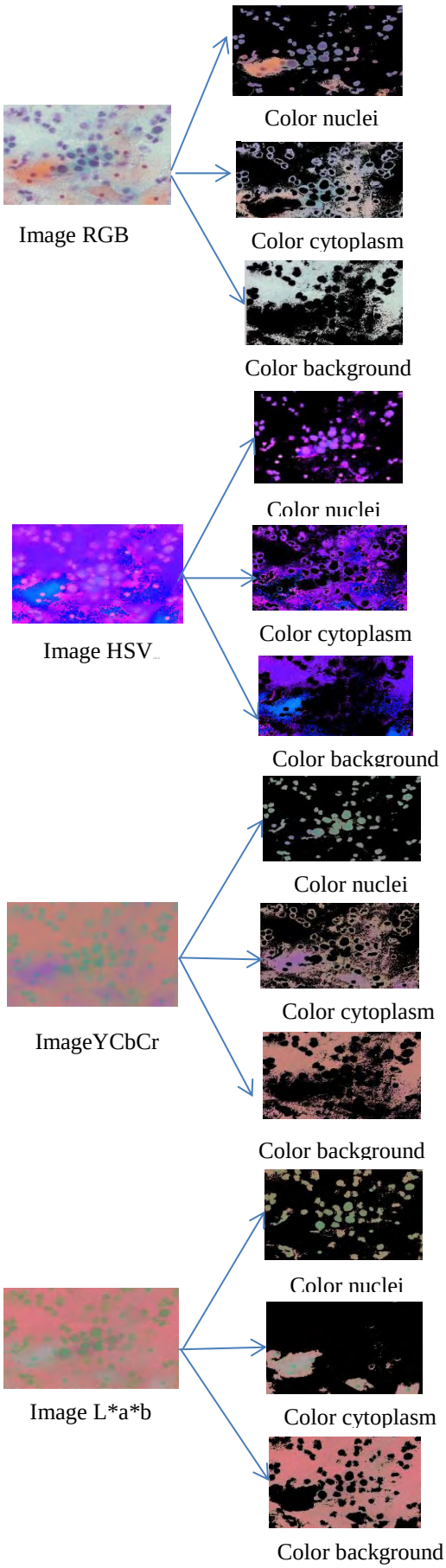


Figure 5 Image Obtained After K -Means Clustering

ISBN 978-80-86943-58-9

4. Conclusion and Future work

the performance of the K-means clustering is evaluated using four different color spaces, RGB, HSV, L*a*b and YCbCr. The experimental results showed that the segmentation results depending on the RGB color space provided the best nucleus segmentation. The average rate of correct segmentation was 64.69% for nucleus segmentation in the RGB color space. This was the best result compared to other color spaces. The average rate of correct cytoplasm segmentation was 81.79%. The present study can be improved by enlarging the dataset and including different kinds of images.

As future work, we will incorporate other heuristics such as size to improve the first phase. We will also experiment with nucleus and cytoplasm classifications using different classifiers.

5. REFERENCES

[BA12] Patel Janak kumar Baldevbhai , R. S. Anand, *Color Image Segmentation for Medical Images using L*a*b* Color Space*, IOSR Journal of Electronics and Communication Engineering, 2278-2834 Volume 1, Issue 2, May-June 2012

[BTP+11]Jorge albertomarcialbasilio , gualbertoaguilartorres, gabrielsánchezpérez, l. Karina toscano medina , héctor m. Pérez meana, Explicit Image Detection using YCbCr Space Color Model as Skin *5th WSEAS international conference on Computer engineering and applications*, 978-960-474-270-7 2011

[B03] E. Bengtsson, Computerized cell image analysis: past, present, and future, in: Proceedings of 13th Scandinavian Conference on Image Analysis, 2003, pp. 395–407.

[CHL+13] Yung-Fu Chen, Po-Chi Huang, Ker-Cheng Lin, Hsuan-Hung Lin, Li-En Wang, Chung Chuan Cheng, Tsung-Po Chen, Yung-Kuan Chan, and John Y. Chiang. Semiautomatic Segmentation and Classificationof Pap Smear Cells. *IEEE journal of biomedical and health informatics, vol. 00, no. 00, 2013*

[DUK08] C. Duanggate , B. Uyyanonvara , and T. Koanantakul. A Review of Image Analysis and Pattern Classification Techniques for Automatic Pap Smear Screening Process. *International Conference on Embedded Systems and Intelligent Technology Bangkok, Thailand*2008

[GP00] A. Garrido*, N. PeHrez de la Blanca. Applying deformable templates for cell image segmentation.ELSEVIER Pattern RecognitionVolume 33, Issue 5, Pages 821–832 2000,

[GK96]Claude Gompel, LeopoldG.Koss, Cytologie Gynécologique et ses bases anatomo-clinique *Paris Pradel 1996*

[KSW07] Kwang-Baek Kim, Doo Heon Song, and Young Woon Woo. Nucleus Segmentation and Recognition of Uterine Cervical Pap-Smears.*Rough Sets, Fuzzy Sets, Data Mining and Granular Computing Springer Berlin Heidelberg.*pp 153-160 2007

[KEHT14] Dina Khattab, HalaMousherEbied, Ashraf Saad Hussein,andMohamedFahmy Tolba1, Color Image Segmentation Based on Different Color Space Models Using Automatic GrabCut, *Hindawi the Scientific World* Journal 2014,

[KSN15] S. Kaaviya, V. Saranyadevi ; M. Nirmala. PAP smear image analysis for cervical cancer detection,*Engineering and Technology (ICETECH), 2015 IEEE International Conference on1 – 4 2015*

[LCC09]Chuen-Horng Lin, Yung-Kuan Chan, Chun-Chieh Chen. Detection and Segmentation of Cervical Cell Cytoplast and Nucleus.*International Journal of Imaging Systems and TechnologyVolume 19, Issue 3, pages 260–270- 2009*

[LEC+98]Olivier Lezoray, AbderrahimElmoataz, Hubert Cardot, Gilles Gougeon, Michel Lecluse, et al..Segmentation of cytological image using color and mathematical morphology.*European conference on Stereology, 1998, Amsterdam, Netherlands.*10 pp, 1998.

[LN07]Zhong Li, KayvanNajarian. biomedical image segmentation based on shape stability. *Image Processing, 2007.ICIP 2007. IEEE International Conference on* VI - 281 - VI – 284-2007

[LH03]N. Lassouaoui, L. Hamami. genetic algorithms and multifractal segmentation of cervical cell images. *Signal Processing and Its Applications, 2003.Proceedings. Seventh International Symposium on1 - 4 vol.2- 2003*

[LEC03]O. Lezoray A. Elmoataz, and H. Cardot1. A color object recognition scheme: application to cellular sorting Machine Vision and Applications 14: 166–171 2003

[MCL91]Michel Mallet, Dominique Chiarasini, Sylvain Labbe, Cytologir Gynécologique Normale Et Pathologique, :*PICCIN 1991*

[MK12]C. MythiliV.Kavitha Color *Image Segmentation using ERKFCM,* International Journal of Computer Applications (0975 – 8887) Volume 41– No.20, March 2012

[M08]Mignotte, M., "*Segmentation by Fusion of Histogram Based -Means Clusters in Different Color Spaces*," in *Image Processing, IEEE Transactions on* , vol.17, no.5, pp.780-787, May 2008

[MKI12]IzzatiMuhimmah, RahadianKurniawan, Indrayanti. Automated Cervical Cell Nuclei Segmentation Using Morphological Operation and Watershed Transformation.*Computational Intelligence and Cybernetics (CyberneticsCom), 2012 IEEE International Conference on* 163 -167 2012

[NAH97] H.Z. Noorani, C. Arratoon, A. Hall, Assessment of Techniques for Cervical Cancer Screening, *Technical Report CCOHTA Report 1997: 2E, Canadian Coordinating Office for Health Technology Assessment* (May 1997)

[NST+02] O Nunobiki, M. sato E. Taniguchi, W. Tang, M. Nakamura,H. Utsunomiya, Y. Nakamura, I. Mori, and K. Kakudo, "Color image analysis of cervical neoplasia using RGB computer color specification,"Anal. Quant. Cytol. Histol., vol. 24, no. 5, pp. 289–294, 2002.[12]

[O14] Adekunle Oguntayo, Cervical Intraepithelial Neoplasia (CIN) (Squamous Dysplasia).*Intraepithelial NeoplasiaInTech* 2014

[OTDC02] F. Ortiz1 , F. Torres1 , E. De Juan2 and N. Cuenca3, Colour Mathematical Morphology For Neural Image Analysis, *Real-Time Imaging* 8, 455–465 2002

[PN13] Marina E. Plissiti,ChristophorosNikou. A Review of Automated Techniques for Cervical Cell Image Analysis and Classification. *Springer Science+Business Media Dordrecht* pp. 1-18, 2013.

[PN11] Marina E. Plissiti, ChristophorosNikou. Automated Detection of Cell Nuclei in Pap Smear Images Using Morphological Reconstruction and Clustering.*IEEE transactions on information technology in biomedicine*, vol. 15, no. 2, 2011

[RKV12] Mr. Vivek Singh Rathore, Mr. MessalaSudhir Kumar , Mr. AshwiniVerma, Colour Based Image Segmentation Using L*A*B* Colour Space Based On Genetic Algorithm, International Journal of Emerging Technology and Advanced Engineering, 2250-2459, Volume 2, Issue 6, 2012

[RNRT12] Rahmadwati; Golshah Naghdy ; Montserrat Ros ; Catherine Todd. Computer aided decision support system for cervical cancer classification. *Proc. SPIE 8499, Applications of Digital Image Processing XXXV, 849919*pp. 1-13, 2012.

[SJ15]Sajeena T JereeshA SAAutomated Cervical Cancer Detection through RGVF segmentation and SVM Classification. IEEE*InternationalConference on Computing and Network Communications (CoCoNet)*663 – 669 2015

A Novel Accurate 3D Surfaces Description Using the Arc-Length Reparametrized Level curves of the Three-Polar Representation

Amal Rihani
CRISTAL Laboratory,
GRIFT research group
ENSI, La Manouba
University
2010, La manouba,
Tunisia
amal.rihani@ensi-uma.tn

Majdi Jribi
CRISTAL Laboratory,
GRIFT research group
ENSI, La Manouba
University
2010, La manouba,
Tunisia
majdi.jribi@ensi.rnu.tn

Faouzi Ghorbel
CRISTAL Laboratory,
GRIFT research group
ENSI, La Manouba
University
2010, La manouba,
Tunisia
faouzi.ghorbel@ensi.rnu.tn

ABSTRACT

This paper studies the problem of the 3D surfaces representation. Our starting point is the extraction of the three-polar representation from the 3D shapes. It consists on a level curves set of the superposition of the three geodesic potentials generated from three reference points of the surface. These curves are characterized by their invariance under the $M(3)$ group of $\mathbb{R}^3$ displacements. We intend to make the arc-length reparametrization of each level curve to ensure its independence to the initial parametrization. The novel representation is materialized by the points of the arc-length reparametrization of all the level curves. Therefore, we obtain an invariant representation under the $M(3)$ transformations group and independent to the initial parametrization. In this work, we implement it on 3D faces since this type of surfaces knows actually a growing interest for the identities determination especially after the many terrorist acts occurred around the world. We experiment, in this context, the identification scenario on a part of the BU-3DFE database. The obtained results show the accuracy of the novel representation.

Keywords

Three-polar, geodesic potential, level set, curve, arc-length, shape representation, invariant, approximation, 3D face, identification.

1 INTRODUCTION

3D shape recognition has become an important issue in the pattern recognition field. This is due especially to the growing development of the 3D scanning tools and the good quality of the obtained 3D data. The pattern recognition with three dimensional data was proposed as an alternative to the one with 2D images. In fact, 3D surfaces permit to overcome the problems of pose and illumination often encountered in 2D data.

However, 3D surfaces lack of a canonical parametrization. Indeed, for the same surface, many parameterizations could exist. They depend on the point of view and the orientation of the surface. This fact makes hard the recognition procedure with 3D data. In order to cross as much as possible these difficulties, the extraction of invariant description from 3D surfaces under some geometrical transformations is proposed as an efficient alternative.

We intend in this work to construct a novel representation of 3D surfaces which is invariant under the $M(3)$ group of transformations ($\mathbb{R}^3$ rotations and translations). This novel representation could be applied to all types of 3D objects. We give, here, a special attention to 3D faces. In fact, this type of surfaces is actually of a paramount importance. It is a powerful tool for the persons identities recognition.

1.1 Related works

We present in this part, an overview of some 3D surfaces description methods including several ones that were implemented on 3D faces. In the literature, the 3D shape description methods can be classified into four main families: the view based methods, the graph based approaches, the global ones and those considered as local.

In the view based approach, a 3D object is characterized by its 2D projections on canonical directions. In fact two objects are assumed to be similar if they are

similar from the same point of view. 2D invariant descriptors could be, then, applied on this set of 2D images in order to extract an accurate representation of the 3D object. The 2D Zernike moments [Che03] and the Fourier descriptors [Vra04] are among of the most used descriptors in this context.

For the graph based methods, we try to represent a 3D object as a graph showing how shape components are linked together. These methods can be classified into two major categories: the Reeb graph one [Tun05] and the Skeleton method [Sun03]. The Reeb graph is a topological structure. It is obtained according to the Morse theory [Shi91] that characterizes a closed 3D surface. In the case of the Skeleton graph, a 3D object is represented by its Skeleton often obtained by the median axis of the used 3D surface.

In the construction of a 3D global shape description method, the representation of a 3D surface is obtained by the geometrical characteristics of the whole object. Several 3D global surfaces description methods were performed in the literature. Osada et al. [Osa02] proposed the 3D distribution forms method. It consists on a novel signature of a 3D object obtained by a probability distribution of a shape function. Paquet et al. [Paq99] were among the first who proposed the famous cords histogram method. It is based on the extraction of the statistical characteristics from the cords of the 3D object. Here, we denote by cords, all segments connecting the gravity center of a 3D object and its triangles centers. The famous 3D Hough Descriptor (3DHD) proposed by Zaharia et al. [Zah01a] is a global descriptor that accumulates the parameters of the representative planes defined by the triangles in a given 3D mesh.

In the fourth approaches category, a local 3D representation is extracted from a 3D surface. Several past 3D local shape description methods were proposed. We mention the pioneer work of Faugeras et al. [Fau86] that characterizes a 3D surface by its high curvature values zones. Zaharia et al. [Zah01b] used the high curvature values surface points or the inflexion ones to extract a statistical description from histograms. Their descriptor is called the shape index histogram. Bannour et al. [Ban00] generalized the idea of the 3D surface description by only the high curvature zones to a method that describes a 3D surface by a set of invariant points corresponding to the levels of the curvature values areas. In the context of 3D faces description with curvatures, Shu-wei et al. [Shu12] used the gaussian curvature to characterize the 3D faces. Here, a 3D face is described by a feature vector of gaussian curvature. The distance between pair of 3D faces is obtained via the distance between their feature vectors. Ganguly et al. [Gan14] proposed to describe a 3D face by a two pairwise curvatures analysis. The first one is the mean, and the maximum curvatures and the second pair corresponds to the gaussian and the minimum curvatures. 3D faces are, then, compared and matched using this description. In order to compare between 3D faces, several methods based on a curvature computation were used to to extract interest points from this type of 3D surfaces. De Giorgis et al. [Deg15] identified fiducial points from 3D faces using a multi-scale curvature analysis. Berreti et al. [Ber13] proposed to use the meshDOG algorithm (Difference Of Gaussian) based on a mean curvature computation to determinate accurate interest points from 3D faces. Another kind of local 3D surface description is based on the geodesic computing around feature points. Many authors [Sam06, Sri08, Gad12] proposed to compute the unipolar representation that consists on the geodesic level curves around a reference point of the 3D shape. They impose, therefore, a coordinates system to the 3D surface. They apply these representations in the context of 3D faces description. Other works, used many unipolar representations around many reference points to locally describe a 3D surface [Maa11]. In order to ensure a more stability of the unipolar representation in the case of error on reference point, Ghorbel et al. [Gho13] proposed a novel representation called the bipolar one. It consists on the invariant set of points corresponding to the levels of the sum of the two geodesic potentials generated from two reference points of the surface. Here, the geodesic information coming from each reference point are combined and not used each one lonely like the representation with many unipolar representations [Maa11]. Jribi et al. [Jri13, Jri14] proposed a novel representation qualified by the three-polar one. It is defined by the invariant points of the surface corresponding to the levels of the sum of the three geodesic potentials generated from three reference points. The same authors proposed an ordered version of the three-polar representation obtained by the intersection between the last one and the radial lines levels representation obtained with the same angular separation [Jri15]. The last representations (bipolar and three-polar) were implemented and tested on some 3D faces.

1.2 Our approach

We propose here a novel 3D surfaces representation. The base of this work is the three-polar one. This last one corresponds to a set of invariant curves under the group of $\mathbb{R}^3$ rotations and translations($M(3)$ group). Once a 3D surface is described by these curves, it becomes more easy to extract an accurate representation from a 3D surface. In fact, the problem of 3D surfaces description is transformed to a problem of 3D curves description. In this context, we try to describe these curves independently to their first parametrizations. We, therefore, characterize them by their arc-length reparametrization. The obtained points from all the curves consist the proposed novel representation. We apply the proposed approach for the description of

3D faces. We use the Hausdorff shape distance as a similarity metric to compare between different shapes.
The reminder of this paper is organized as follows: We detail in the second section all the steps of the novel representation construction. In the third section, we expose the used similarity metric that corresponds to the Hausdorff Shape distance. We apply, finally, in the fourth section the novel representation for the description of 3D faces. The obtained results for the identification scenario on a part of the BU-3DFE database [Lij06] of 3D faces are exposed.

2 CONSTRUCTION OF THE NOVEL 3D SURFACES REPRESENTATION

We intend in this work to describe a 3D surface by an accurate, finite and invariant set of points under the geometrical transformations of the $M(3)$ group. We suppose here that a 3D object is a continuous surface. It is considered as a 2D-differential manifold that we denote by S. The three-polar representation, known by its stability under the errors on the reference points extraction [Jri14] is used as a starting representation. This last one corresponds to a set of invariant curves under the same group of transformations. Finite and accurate points are obtained by the discretization of each curve. The discretization procedure has a major importance for the construction of the novel representation. In fact, an accurate discrete representation of the level curves leads to an efficient novel representation. Therefore, the steps of the novel 3D representation can be summarized as follows: (i) The first step consists on the three-polar representation construction. (ii) In the second step, an accurate description of each three-polar level curve should be performed.
We use the following mathematical considerations for the construction of the novel representation. Let P_1 and P_2 be two points of S. We denote by:

- $\gamma(P_1,P_2)$: the geodesic curve joining P_1 and P_2. It is the curve having the minimum of distance between P_1 and P_2 and belonging to the surface S.
- $\widetilde{\gamma}(P_1,P_2)$: the length of the geodesic curve computed between P_1 and P_2
- $U_r(P)$: the geodesic potential generated from a point r of S. It is the function that computes for each point P of S the length of the geodesic curve joining it to the point r.
  We describe in the rest of the section the two steps cited above.

2.1 Brief recall of the three-polar representation

The three-polar representation is constructed from three reference points of a 3D surface S. It is built in order to ensure a more stability in the case of extraction errors on the reference points [Jri14]. This 3D representation consists on a set of curves extracted from the 3D object assumed to be, here, a 2D-differential manifold. These curves correspond to the levels of the sum of the three geodesic potentials generated from the used three reference points of the surface. It is easy to see that these level curves are invariant under the geometrical transformations of the $M(3)$ group since the geodesic computation is invariant under the same transformations.
Therefore, let denote by P_1, P_2 and P_3 three reference points of S, U_{P1}, U_{P2} and U_{P3} their corresponding geodesic potential functions and U_3 the sum of these three geodesic potentials.
The three-polar representation composed by a set of K level curves can be formulated as follows:

$$M^k(S) = \{C^{\lambda_i}\}_{i=1..k} \tag{1}$$

where C^{λ_i} is the level curve with the value λ_i of the sum U_3 of the three geodesic potentials generated from the used three reference points. Therefore:

$$C^{\lambda_i} = \{p \in S, U_3(p) = \lambda_i\} \tag{2}$$

We note here that the curves $\{C^{\lambda_i}\}_{i=1..k}$ are extracted from the 3D surface with the same step of the sum of the three geodesic potentials.

2.2 Accurate description of the level curves

A 3D object is assumed to be a 2D-differential manifold. It is represented by a collection of indexed 3D curves $\{C^{\lambda_i}\}_{i=1..k}$ of the three-polar representation. A level curve C^{λ_i} parametrization denoted by $C^{\lambda_i}(t)$ is a 1-periodic function of a continuous parameter t defined by:

$$C^{\lambda_i}(t) : [0,1] \to \mathbb{R}^3 \tag{3}$$
$$t \mapsto [x(t), y(t), z(t)]^t$$

It is important to note that for the same curve we can find many parametrizations. They depend on the position and the orientation of the used curve and the speed we go over it. This fact makes hard the comparison between curves. In order to overcome this problem, we propose to use a $\mathbb{G}$-invariant reparametrization of each curve. $\mathbb{G}$ is group of the geometrical transformations applied to a curve. A reparametrization of $C^{\lambda_i}(t)$, noted $C^{\lambda_i}(\widehat{t})$, is defined as follows :

$$C^{\lambda_i}(\widehat{t}) = C^{\lambda_i}(\tau(t)) = [x(\tau(t)), y(\tau(t)), z(\tau(t))]^t, t \in [0,1] \tag{4}$$

where τ is an increasing function defined on [0,1].
Let consider $C_1^{\lambda_i}(t_1)$ and $C_2^{\lambda_i}(t_2)$ two parameterizations of a curve C^{λ_i} and its image by the geometrical transformation $g \in \mathbb{G}$. After the $\mathbb{G}$-invariant reparametrization,

we obtain:

$$C_2^{\lambda_i}(\widehat{t}) = g(C_1^{\lambda_i}(\widehat{t}+t_0)) \quad (5)$$

where $t_0 \in \mathbb{Z}$, $g \in \mathbb{G}$ and t_0 is the starting points difference between the curves.
In our context, $\mathbb{G}$ corresponds to the $M(3)$ group formed by the $\mathbb{R}^3$ rotations and translations. This transformations group preserves the length of curves. The speed we go over a curve will affect the parametrization. We perform, therefore, the arc-length reparametrization of this curve. This implies that it is covered with a constant speed. The arc-length reparametrization of a 3D curve C^{λ_i} is defined as follows:

$$S(t) = 1/L \int_0^t \sqrt{x(t)'^2 + y(t)'^2 + z(t)'^2} dt, t \in [0,T] \quad (6)$$

Here, L denotes the length of the level curve C^{λ_i}.

3 SIMILARITY METRIC

In order to compare between 3D shapes, we use the novel 3D representation as a signature. The well known Hausdorff shape distance introduced by Ghorbel et al. [Gho98, Gho12] is used as a similarity metric. Let G be the group of all possible parameterizations of surfaces. It can be the $\mathbb{R}^2$ plane for the open surface or the $\mathbb{S}^2$ for the closed ones. In the context of 3D surfaces pieces diffeomorphic to G, on which act the $M(3)$ group, the Hausdorff shape distance can be defined for two surfaces pieces S_1 and S_2 and two displacements g_1 and g_2 as follows:

$$\triangle(S_1,S_2) = max(\rho(S_1,S_2),\rho(S_2,S_1)) \quad (7)$$

where:

$$\rho(S_1,S_2) = \sup_{g_1 \in M(3)} \inf_{g_2 \in M(3)} \|g_1 S_1 - g_2 S_2\|_{L^2}^2 \quad (8)$$

Since the $M(3)$ displacement group preserves this norm, the Hausdorff shape distance can be reduced to the following quantity:

$$\triangle(S_1,S_2) = \inf_{h \in M(3)} \|S_1 - hS_2\|_{L^2}^2 \quad (9)$$

In order to compute the Hausdorff shape distance value between two surfaces, the optimal transformation between these two objects should be determined. We use in this context, The Iterative Closest Point (ICP) algorithm [Bes92] to estimate this transformation and thus to reach the real value of this distance.

4 DESCRIPTION OF 3D FACES WITH THE NOVEL REPRESENTATION

Actually, human recognition via biometric traits is of a paramount importance especially with the many terrorist acts occurred around the world. The face is one of the most used biometric traits since it does not require the cooperation of the subjects. The 3D faces description is becoming actually an area of growing interest especially with the rapid development of 3D scanning tools. We try, in this context, to apply the novel representation for the description of 3D faces. We present in the rest of this section the used database and we detail the construction steps of the novel representation on this special type of 3D surface.

4.1 The used database

In order to study the performance of the novel 3D representation for the description of 3D faces, we use the BU-3DFE database [Lij06] which contains 100 subjects (56 females and 44 males) from different ethnicities. Seven facial expressions (neutral, disgust, happiness, angry, surprise, sadness and fear) are available for each subject. Each facial expression is presented by four levels of magnitude.

4.2 The used three reference points

The selection of the reference points is the first step of the three-polar representation construction. The nose tip which is used in the unipolar representation based on only one reference point [Sam06, Sri08, Gad12] will be also used as a reference point for the three-polar representation. The two outer corners of eyes will be selected as candidate reference points. For the automatic extraction of the reference points, we use the approach proposed by Szeptycki et al. [Sze09a] which is based on a curvature analysis of a 3D face.

4.3 Geodesic computation

We have assumed in the construction of the three-polar representation that a 3D surface is a 2D-differential manifold. In practice, this surface is represented by a 3D mesh composed by a set of vertices and edges.
In order to compute the geodesic potentials from the reference points, we should be able to compute the geodesic curves between pairs of points of the 3D discrete mesh. We use in our work the fast marching algorithm [Set96] to compute geodesic paths between pairs of points and subsequently the geodesic potentials.

4.4 Extraction of the level curves of the three-polar representation

Since the 3D face corresponds to a discrete object, its level curves of the three geodesic potentials sum are also discrete. Each level curve will be composed by a set of points from the 3D face. In practice a level curve of value λ can be seen as a trip. It is formulated as follows:

$$C^{\lambda} = \{P \in S, \lambda - \varepsilon \leq U_3(P) \leq \lambda + \varepsilon\} \quad (10)$$

where ε is a real positive value chosen according to the resolution of the mesh to avoid the intersections between successive level curves.

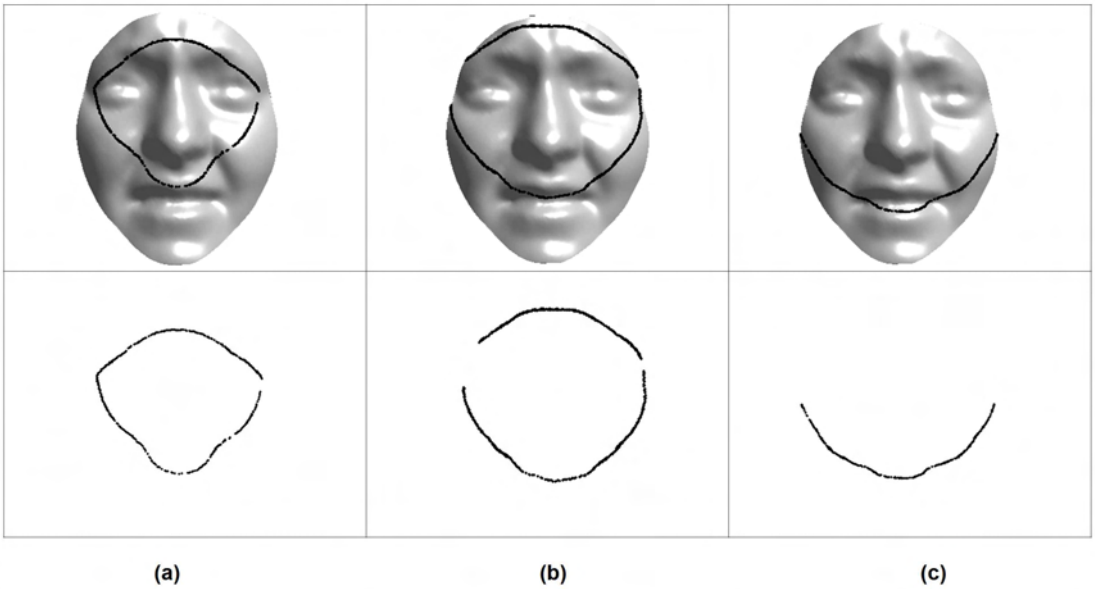


Figure 1: Different kinds of level curves from the three-polar representation. (a): A closed level curve. (b): An open level curve with two separated parts. (c): An open level curve composed by one part.

4.5 Arc-length reparametrization of the discrete level curves

After the extraction of the discrete level curves from a 3D face, we proceed to their arc-length reparametrization. In the context of 3D faces study, the 3D surfaces are open. Therefore, it is naturally to obtain some open level curves since we reach the surface border. Fig. 1 illustrates many kinds of obtained level curves from the three-polar representation. Fig. 1(a) shows an example of a closed level curves. Fig. 1(b) illustrates an open level curves composed by two separated parts. The curve of the Fig. 1(c) corresponds to an open curve with only one part.

In order to make all curves closed for the computation of the arc-length reparametrization, we propose in this work to complete the empty parts of the open level curves by some border points. Fig. 2 illustrates some open curves that were completed by the used border points of the 3D surface. Once all the level curves are closed, before we perform their arc-length reparametrization, we approximate each one of them by the B-spline function. Let $\{p_{i,\lambda}\}_{i=0..N_\lambda}$ be the set of the N_λ discrete points of a curve C^λ of level value equal to λ. The approximated curve by the B-spline function denoted by $C^\lambda(t)$ can be formulated as follows:

$$C^\lambda(t) = \sum_{i=1}^{N_\lambda} B_{i,k-1}(t) \times ((1 - \frac{t-t_i}{t_{i+k}-t_i})P_{i-1,\lambda} + \frac{t-t_i}{t_{i+k}-t_i}P_{i,\lambda}) \quad (11)$$

where $B_{i,k}(t)$ is the B-spline basic functions.
We apply, then, the arc-length reparametrization of each approximated level curve. We obtain equidistant points on each curve since we go over it with the same speed. Fig. 3(a) represents a level curve of the three-polar representation extracted from a 3D face. Fig. 3(b) shows the approximation of the same curve by the B-spline function and Fig. 3 (c) illustrates the arc-length reparametrization of this curve.

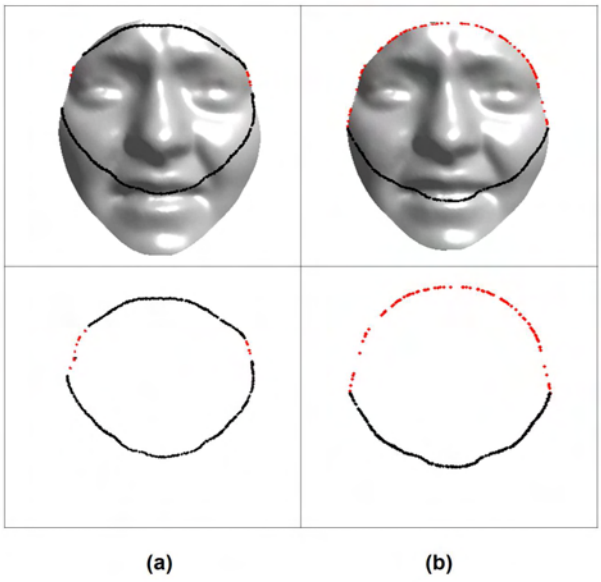


Figure 2: Illustration of two open level curves completed by the border points.(a): A curve with two separated parts. (b): A curve with one part.

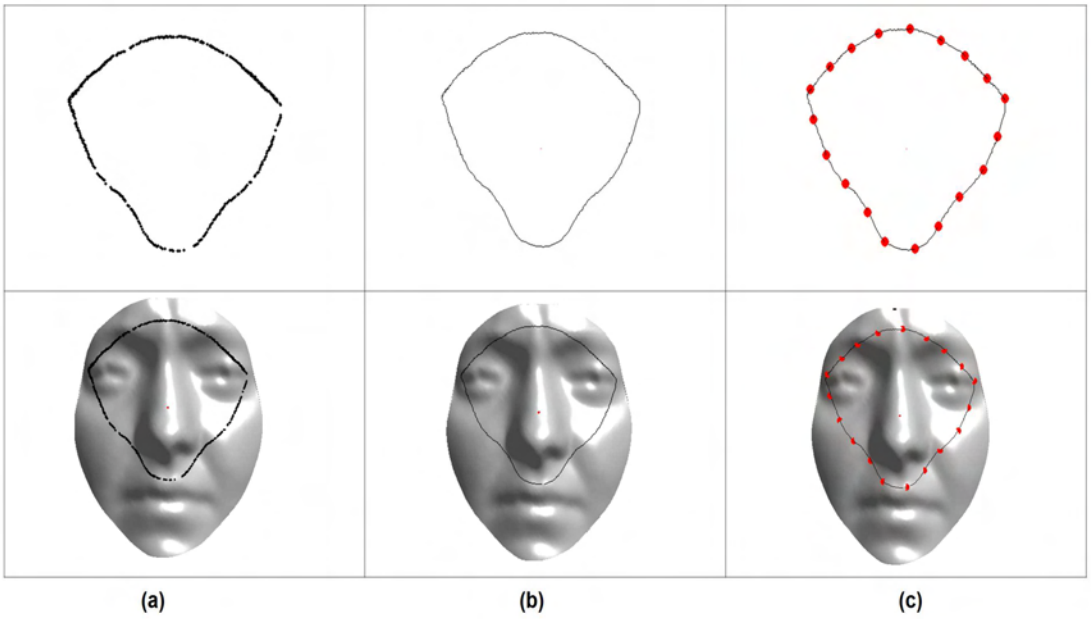


Figure 3: The steps of the description of a level curve from the extraction to the arc-length reparametrization. (a): A discrete level curve of the three-polar representation. (b): Approximation with the B-spline function. (c): The obtained points after the arc-length reparametrization.

4.6 Accuracy of the novel representation for the description of 3D faces

We test, here, the performance of the novel representation for the description of the 3D faces. We use for the experimentation a part of the database BU-3DFE [Lij06]. This portion is composed by the first magnitude level of each facial expression and the neutral face of all the database subjects (100 persons). A total of 700 faces are, then, used for the experimentation.
Fig. 4 presents all the steps of the novel representation construction for two faces with different facial expressions going from the three-polar representation extraction (Fig. 4(a)) to the construction of the novel representation (Fig. 4(d)).
We focus our study on the identification scenario. In this case, a person is compared to all the individuals of the used database and matched to the most similar ones. We run the experiments with the protocol All vs All which consists on the comparison of each face of the database to all the others. Here, the gallery subset and the probes set corresponds to the all 700 faces.

Fig. 5 shows the Cumulative Matching Curve of the proposed 3D representation under the protocol All vs. All. The obtained results are about 92.68% for the rank one recognition rate. This significant recognition rate proves the accuracy of the novel 3D surfaces representation for the description of 3D faces.

5 CONCLUSION

We introduced in this work a novel 3D surfaces description, which is based on the three-polar representation. This last one consists on a set of invariant curves under the $M(3)$ group of $\mathbb{R}^3$ rotations and translations. These curves correspond to the levels of the superposition of the three geodesic potentials generated from three reference points of the surface. The novelty of this work lies on the arc-length reparametrization of each curve of the three-polar representation. This fact makes them independent to the initial parametrization. We apply the novel representation for the description of 3D faces knowing actually a growing interest for the determination of persons identities. To illustrate the performance of the proposed representation for 3D faces description, we implement, in this context, the identification scenario on a part of the BU-3DFE database. We obtain a rank one recognition rate of about 92.68% .
We propose in future work to experiment the novel representation on the standard database of 3D faces FRGC V2. An important perspective is to make a study for the best choice of the three reference points for the three-polar representation. We intend also to compare the proposed representation with some works of the state of the arts using the standard protocols.

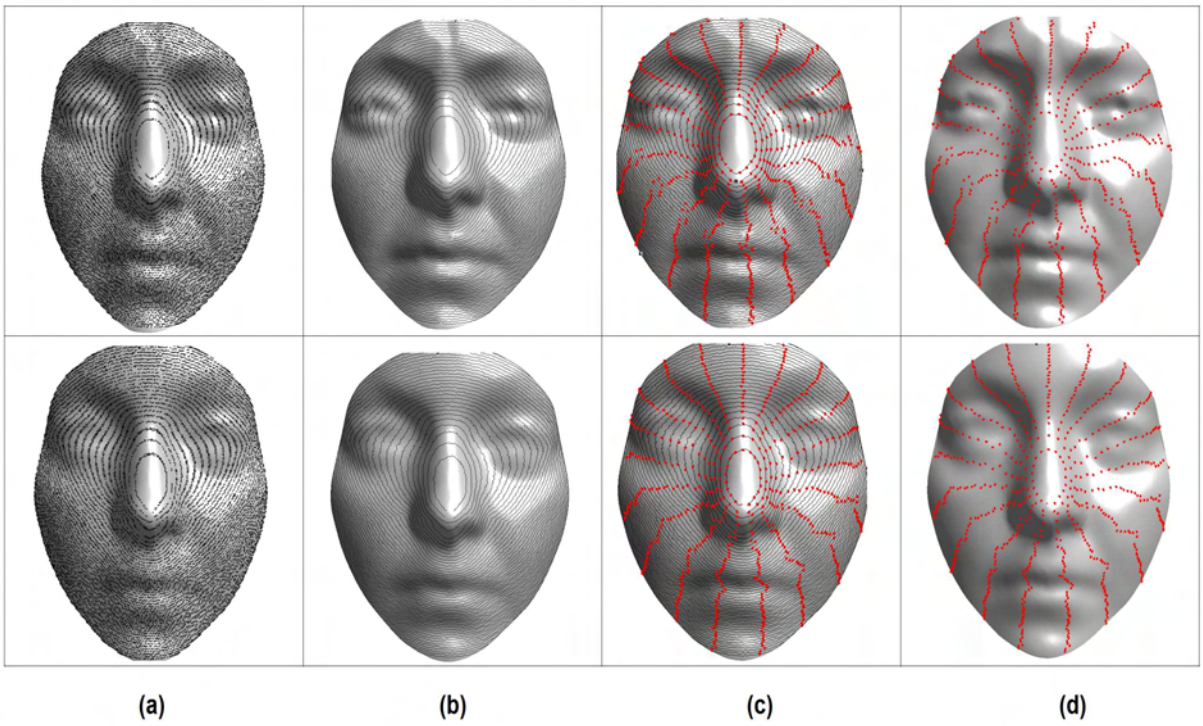


Figure 4: (a): The level curves of the three-polar representation. (b): Approximation of the level curves with the B-spline function. (c): The arc-length reparametrization of the level curves. (d) The obtained points of the novel 3D representation.

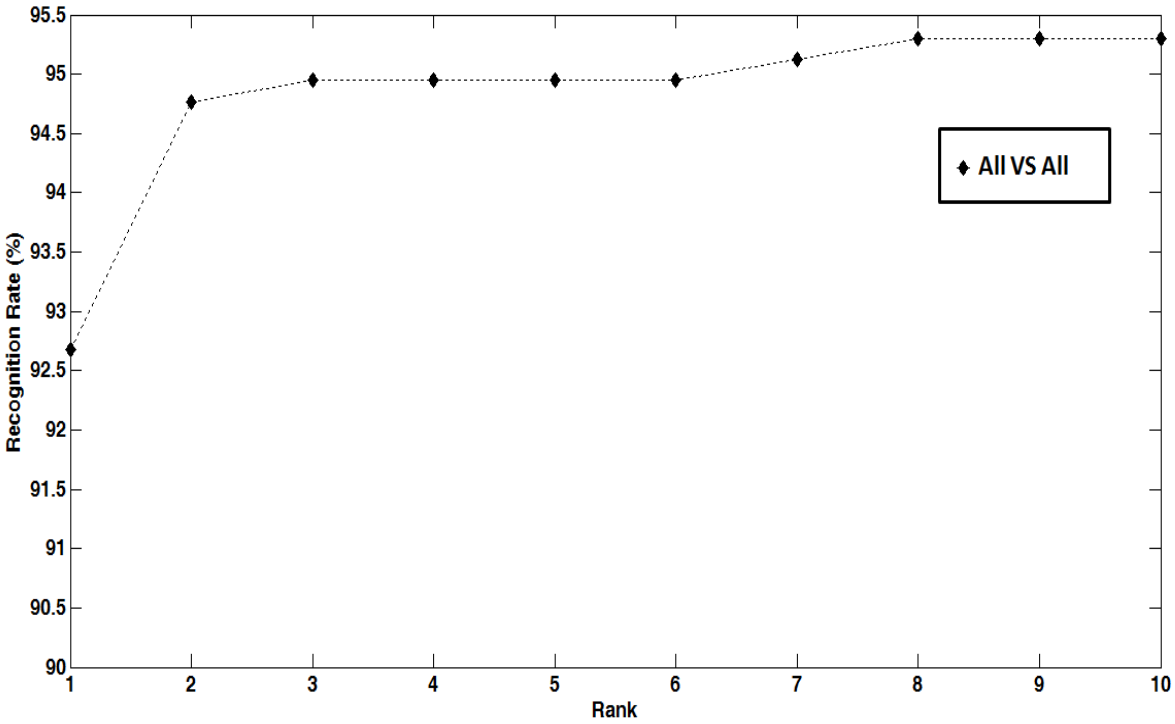


Figure 5: The CMC curve of the proposed approach for the scenario: All vs. All.

6 REFERENCES

[Ban00] Bannour, M.T., and Ghorbel, F. Isotropie de la représentation des surfaces; Application à la description et la visualisation d'objets 3D, In Proc. RFIA 2000, pp. 275-282, 2000.

[Ber13] Berretti, S., Werghi, N., Del Binbo, A., and Pala, P. Matching 3D face scans using interest points and local histogram descriptors, Journal of Computers and Graphics, vol. 37, No 5, pp. 509-525, 2013.

[Bes92] Besl,P.J., and Mckay, N.D. A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, No 2, pp. 239-256, 1992.

[Che03] Chen, D.Y., Tian, X.P., Shen, Y.T., and Ouhyoung, M. On Visual Similarity Based 3D Model Retrieval, Computer Graphics Forum, vol. 22, No 3, pp. 223-232, 2003.

[Deg15] De Giorgis, N., Rocca, L., and Puppo, P. Scale-space techniques for fiducial points extraction from 3D faces, In Proc. ICIAP'15, the 18^{th} International Conference on Image Analysis and Processing, pp. 421 - 431, 2015.

[Fau86] Faugeras, O.D., and Hebert, M. The representation, recognition and positioning of 3D shapes from range data, techniques for 3D machine perception, Edition A, Rosenfield, Hollande, 1986.

[Gad12] Gadacha, W., and Ghorbel, F. A new 3D surface registration approach depending on a suited resolution: Application to 3D faces, In Proc. MELECON'12, the IEEE Mediterranean and Electrotechnical Conference, 2012.

[Gan14] Ganguly, S., Bhattacharjee, D., and Nasipuri, M. 3D face recognition from range images based on curvature analysis, ICTACT Journal on Image and Video Processing, vol. 4, No 3, pp. 748, 2014.

[Gho98] Ghorbel, F. A unitary formulation for invariant image description: application to image coding, Springer, Annals of telecommunications, vol. 53, No 5-6, pp. 242-260, 1998.

[Gho12] Ghorbel, F. Invariants for shapes and movement. Eleven cases from 1D to 4D and from euclidean to projectives (French version), Arts-pi Edition, Tunisia, 2012.

[Gho13] Ghorbel, F. and Jribi, M. A robust invariant bipolar representation for R3 surfaces: applied to the face description, Springer, Annals of telecommunications, vol. 68, No 3-4, pp. 219-230, 2013.

[Jri13] Jribi, M., and Ghorbel, F. An Invariant Three-polar Representation for R^3 Surfaces: Robustness and Accuracy for 3D Faces Description, In Proc. SCSI'13, the International Conference on Systems, Control, Signal Processing and Informatics, 2013.

[Jri14] Jribi, M., and Ghorbel, F. A Stable and Invariant Three-polar Surface Representation: Application to 3D Face Description, In Proc. WSCG'14, the 22^{nd} International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2014.

[Jri15] Jribi, M., and Ghorbel, F. A Geodesic Based Approach for an Accurate and Invariant 3D Surfaces Representation, In Proc. WSCG'15, the 23^{rd} International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2015.

[Kan93] Kang, S.B., and Ikeuchi, K. The Complex EGI: A New Representation for 3D Pose Determination, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, No 7, pp. 707-721, 1993.

[Lij06] Lijun, Y., Xiaozhou, W., Yi, S., Jun, W., and Matthew, J. A 3D Facial Expression Database For Facial Behavior Research, In Proc. FG'06, the 7^{th} International Conference on Automatic Face and Gesture Recognition, pp. 211 - 216, 2006.

[Maa11] Maalej, A., Ben Amor, B., Daoudi, M., Srivastava, A., and Berretti, S. Shape analysis of local facial patches for 3D facial expression recognition, IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 44, No 8, pp. 1581-1589, 2011.

[Osa02] Osada R., Funkhouser T., Chazelle B., Dobkin D.: Shape distributions. ACM Transactions on Graphics, pp. 807-832, 2002.

[Paq99] Paquet E., Rioux M. A query by content system for three-dimensional model and image databases management. In Proc. the 17^{th} conference on Image and Vision Computing, pp. 157-166, 1999.

[Sam06] Samir, C., Srivastava, A., and Daoudi, M. Three dimensional face recognition using shapes of facial curves , IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, No 11, pp. 1858-1863, 2006.

[Set96] Sethian, J, A., A Fast Marching Level Set Method for Monotonically advancing fronts, In Proc.Nat.Acad.Sci., 1996.

[Shi91] Shinagawa Y., Kunii T.-L., Kergosien Y.-L. Surface coding based on morse theory. In Proc. IEEE Comput. Graph. Appl. 11, pp. 66-78, 1991.

[Shu12] Shu-Wei L., Shu-Shen H., Jui-Lun Ch., Sheng-Yi Li.3D Face Recognition Based on Curvature Feature Matching with Expression Variation, Journal of Advances in Intelligent Systems and Computing, vol. 193, pp. 289-299, 2012.

[Sri08] Srivastava, A., Samir, C., Joshi, S.H., and Daoudi, M. Elastic shape models for face anlysis using curvilinear coordinates, Journal of Mathematical Imaging and Vision, vol. 33, No 2, pp. 253-265, 2008.

[Sun03] Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton based Shape Matching and Retrieval, Shape Modeling International 2003, p. 130, 2003.

[Sze09a] Szeptycki, P., Ardabilian, M., and Chen, L. A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking, In Proc. BTAS'09, the IEEE 3^{rd} International Conference on Biometrics: Theory, Applications, and Systems, 2009.

[Tun05] Tung, T., and Schmitt, F. The Augmented Multiresolution Reeb Graph Approach for Content-Based Retrieval of 3D Shapes, International Journal of Shape Modeling, vol. 11, No 1, pp.

[Vra04] Vranic, D.V. 3D Model Retrieval.PhD dissertation, University Of Leipzig, 2004. 91-120, 2005.

[Zah01a] Zaharia, T., and Preteux, F. Hough transform-based 3D mesh retrieval. In Proc. the 10^{th} SPIE Conference on Vision Geometry, pp. 175-185, 2001.

[Zah01b] Zaharia, T., and Preteux, F. 3D shape-based retrieval within the MPEG-7 framework. In Proc. the 10^{th} SPIE Conference on Vision Geometry, pp. 133-145, 2001.

Integration of statistical spatial relations into Active Shape Model- Application to striatum segmentation in MRI

Saïd Ettaïeb
Research laboratory of signal, image and information Technology - University of Tunis El Manar-Tunisia
Rommana 1068, Tunis-B.P. n° 94, Tunisia
settaieb@gmail.com

Besma Mnassri
Higher Institute of Applied Sciences and Technology- University of Gafsa-Tunisia
Campus Universitaire-Sidi Ahmed Zarrouk – 2112, Gafsa, Tunisia
mnassribesma9@gmail.com

Kamel Hamrouni
Research laboratory of signal, image and information Technology - University of Tunis El Manar-Tunisia
Rommana 1068, Tunis-B.P. n° 94, Tunisia
kamel.hamrouni@enit.rnu.tn

ABSTRACT

This paper describes a new method based on Active Shape Model (ASM) and statistical spatial relations. It combines three types of a priori knowledge: the structures shapes, the distance and the angle variability between them. This knowledge is estimated during a training step. Then, the obtained models are used to guide the evolution of initial shapes during the segmentation step. The proposed method is applied to extract the striatum (Caudate nucleus and Putamen) on MR images of the brain. The obtained results are promising and show the performance of the proposed method.

Keywords

Statistical a prior knowledge, spatial relations, active shape model, MRI.

1. INTRODUCTION

Segmentation of medical images is a major issue and one of the most challenging topics. However, it is a hard task because of many factors. Indeed, medical images (scintigraphy, MRI, scanner...) are often characterized by low contrast, low resolution and the presence of noise. Moreover, the anatomical structures to be extracted are always complex and variable.

The conventional methods based only on the low-level characteristics of the image are not reliable, because the intensity of a pixel cannot guarantee an effective segmentation. To overcome these limits, many recent methods are proposed. They are taking into account high-level a priori knowledge, related to the anatomical structures during segmentation such as shape, texture, position, etc. These methods provide a powerful solution for a robust segmentation.

Among a priori knowledge, we can cite the spatial relations between structures which are often more stable than the appearance characteristics of the structures themselves. In this context, we proposed to integrate spatial relations into active shape model-ASM [Coo95]. The main idea is to exploit a priori knowledge of shape that exists in ASM and introduce new a priori knowledge about distance and angle variation between structures to be segment.

The aim is to define a new robust method well adapted for the segmentation of two structures, using three types of statistical a priori knowledge: the shape of each structure, the distance and the orientation variability between them. This knowledge is modeled during a training step, then, the obtained models are used to guide the segmentation process and guarantee the preservation of the distance and angle between shapes in the authorized intervals.

The proposed method is validated on a clinical application, where the problem consists in segmenting two structures of interest: caudate nucleus and putamen on MRI slices of the brain.

This paper is organized as follows: Section 2 reviews briefly related work. Section 3 is devoted to the integration of statistical spatial relations to guide the segmentation process. Finally, in Section 4, the proposed model is applied to localize two internal brain structures on MRI slices (caudate nucleus and putamen). The work is concluded in Section 5.

2. RELATED WORK

Many approaches for medical images segmentation have been developed over the years based on several techniques. First, conventional methods do not use any a priori knowledge and are fully based on low-level features mainly pixels intensities. The main drawback of these methods is being not robust enough because sometimes intensities in the same tissue are heterogeneous.

Such methods are highly sensitive to noise and produce satisfactory results unless if the contrast between structures is sufficiently marked.

To overcome these limits, new approaches based on a priori knowledge have been proposed. Among these methods, deformable models are widespread. They based on a priori knowledge of shape. They consist to put a curve close to the structure to be extracted that will be moved progressively to coincide to the edges of the region of interest while minimizing an energy term.

In this work, we are interested to these approaches because their principle is general and flexible making possible the integration a priori knowledge such as the spatial relations. Indeed, in literature, three basic types of spatial relations can exist between objects in an image: topological relations and metric relations who are in turn are partitioned to distance relations and direction relations [Hud08]. The topological relations represent the adjacency between structures. They show how an object partially or completely covers another object ("is adjacent to", "crosses ", "is included"). The distance relations describe the distance between structures ("close", "far", "to a distance of ") and the direction relations based on the six usual directions.

In the medical context, among the first remarkable work using spatial relations, we find that of Perchant [Per02]. He proposes a brain structures recognition procedure based on the matching of graphs: a graph derived from a reference image manually segmented by an expert and a graph of the image to be recognized. In [Gér00] Géraud et al. have proposed a sequential method of recognition of brain structures, where each structure is recognized through the structural information resulted from previously recognized structures. This information is generated from relations of distance and direction defined with respect to the already segmented structures.

However, in these works, spatial relations are always used in the recognition step, whereas the segmentation was achieved with conventional methods. To relieve these drawbacks, Colliot [Col04] invented a new methodology, which consists to directly introduce spatial relations in segmentation step. The segmentation is realized from the beginning in a region of interest defined by spatial relations. The spatial relations (direction, distance and adjacency) are represented by fuzzy sets and incorporated into the evolution equation of the active contour [Kas87a] as an external force. For the segmentation of a given structure, this force attracts the curve to the image areas where the spatial relationships are considered verified. The segmentation process is sequential. It is based on a graph that describes, in a hierarchical manner, the spatial relationships of brain anatomy.

Other recent works are published [Nem09, Fou10] where spatial relations are used either in the recognition step or in the segmentation step.

However, few works have opted for the integration of spatial knowledge into active shape models. One example is the work of Barhoumi et al. [Bar15] who proposed to incorporate a spatial relation of direction into an active shape model for the detection of Region of Interest in medical images. This spatial relation is modeled using fuzzy membership functions in order to model the uncertainty and the ambiguity of the spatial representation. In the same context, in [Jaa11], the authors have introduced a method that consists to add a spatial relation of distance to the active shape model. The a priori knowledge of spatial relation stems from a fuzzy logic modeling phase. In [Ett14], Ettaïeb et al. introduced a new statistical model of shape and spatial relation based on a priori knowledge of shape and a priori knowledge about the variation of a spatial distance relation.

The above methods have remarkably performed the medical images segmentation. Nevertheless, they have some known limitations. Indeed, the majority of them have combined a priori knowledge of shape which exists in the active shape model with one extra constraint either of distance or direction. In the present work, we propose to integrate two types of spatial relations into active shape model: spatial distance relation "A is at a distance of B" and spatial orientation relation based on the angle variation between two structures. These relations will be modeled statistically in a training step and used directly in the segmentation procedure.

3. ACTIVE SHAPE MODEL INTEGRATING STATISTICAL DISTANCE AND ORIENTATION MODELS

The basic idea of our contribution is to exploit a priori knowledge of shape that exists in ASM and introduce new a priori knowledge about distance and angle variation between the structures to be segment. This new knowledge will be estimated during a training step by two models: a distance model and an orientation model. These models will be then used to constrain the evolution of the shapes to the target structures and ensure maintenance of the distance and the angle between structures in the allowed intervals.

Thus, the proposed method requires two main steps:

• A training step, which aims to deduce, from a set of sample images, four basic models: a statistical shape model for each structure, a statistical distance model and a statistical orientation model.

• A segmentation step, based on the obtained models to guide the evolution of two initial shapes to the target structures.

Training Step

This step consists in collecting at first a set of samples of images reflecting the possible variations of two structures to be segmented. Then, we extract, from each image, the shape of each structure by placing a sufficient number of landmarks on the target contours. Considering that *n* and *m* are respectively the number of landmarks required to represent the details of the first and the second structure and *N* is the number of images in the training set, each structure can be represented by a matrix of points defined as follows:

$M_{st_1}(\boldsymbol{2n}, N) =$

$\boldsymbol{v_{11}}$	$\boldsymbol{v_{21}}$	$\boldsymbol{\dots v_{i1} \dots}$	$\boldsymbol{v_{N1}}$
x_{111}	x_{211}	….	x_{N11}
y_{111}	y_{211}	….	y_{N11}
⋮	⋮	….	⋮
x_{11n}	x_{21n}	….	x_{N1n}
y_{11n}	y_{21n}	….	y_{N1n}

$M_{str_2}(\boldsymbol{2m}, N) =$

$\boldsymbol{v_{12}}$	$\boldsymbol{v_{22}}$	$\boldsymbol{\dots v_{i2} \dots}$	$\boldsymbol{v_{N2}}$
x_{121}	x_{221}	….	x_{N21}
y_{121}	y_{221}	….	y_{N21}
⋮	⋮	….	⋮
x_{12m}	x_{22m}	….	x_{N2m}
y_{12m}	y_{22m}	….	y_{N2m}

With v_{ij} is the vector of points which models the structure *j* on the image *i*. (x_{ijk}, y_{ijk}) are the coordinates of the point *k* placed in the image *i* on the contour of the structure *j*. From these two matrices, the shape model of each structure and the corresponding distance and orientation models can be constructed. Indeed, from two matrices of points obtained, we can calculate the mean shape relative to each structure [Ham98]:

$$\bar{V}_1 = \frac{1}{N}\sum_{i=1}^{N} v_{i1} \quad (1)$$

$$\bar{V}_2 = \frac{1}{N}\sum_{i=1}^{N} v_{i2} \quad (2)$$

Then, we can determine the modes and the amplitudes of deformation of every shape by applying the PCA on aligned shapes. Each structure can be represented by a shape model that describes its geometry and deformation modes. These models can be respectively defined by Equations (3) and (4). They represent a priori knowledge of shape of each structure.

$$V_1 = \bar{V}_1 + P_1 b_1, \quad (3)$$

$$V_2 = \bar{V}_2 + P_2 b_2, \quad (4)$$

With: P_1 and P_2 are respectively the matrices of the main deformation modes of the first and the second structure. b_1 and b_2 are two weight matrices which represent respectively the projection of the shape V_1 in the base P_1 and the shape V_2 in the base P_2

3.1.1 Construction of the Statistical Distance Model

The statistical distance model is made at the same time as that of the shape's models. It first consists in computing the distances between both structures of interest from the training images and then trying to deduce a compact and precise formulation, which describes the authorized distances. Given an image *i* of the training set where both structures of interest are modeled respectively by the two following vectors:

$$v_{i1} = (x_{i11}, y_{i11}, \dots, x_{i1j}, y_{i1j}, \dots, x_{i1n}, y_{i1n}) \quad (5)$$

$$v_{i2} = (x_{i21}, y_{i21} \dots, x_{i2k}, y_{i2k}, \dots, x_{i2m}, y_{i2m}) \quad (6)$$

First, we proceed to calculate the centers of gravity of two structures: $B_{i1}(Gx_{i1}, Gy_{i1})$ and $B_{i2}(Gx_{i2}, Gy_{i2})$. For example, the calculation of center of gravity of a structure modeled by a vector v_{i1} is as follows [Bou88]:

• Surface of the structure:

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \quad (7)$$

• Coordinates of center of gravity:

$$G_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (8)$$

$$G_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (9)$$

Then, the Euclidean distance between B_{i1} and B_{i2} is defined by:

$$d(B_{i1}, B_{i2}) = \sqrt{(Gx_{i1} - Gx_{i2})^2 + (Gy_{i1} - Gy_{i2})^2} \quad (10)$$

Therefore, the elementary distance d_i between the two structures of interest in an image *i* can be defined by:

$$d_i(v_{i1}, v_{i2}) = d_i(v_{i2}, v_{i1}) = d(B_{i1}, B_{i2}) \quad (11)$$

With the same principle, we can calculate the distances between the two structures of interest through all the images of the training set. Thereby obtaining a vector of distances of dimension *N*:

$$v_d = (d_1, d_2, \dots, d_i, \dots, d_N) \quad (12)$$

The objective now is to deduce a compact formulation that describes authorized distances. Indeed, from the vector v_d, we can calculate the following basic statistical parameters:

- The mean distance between two structures of interest:

$$d_m = \frac{1}{N}\sum_{i=1}^{N} d_i \quad (13)$$

- The variance which measures the dispersion of elementary distances d_i around the mean distance:

$$V(v_d) = \frac{1}{N} \sum_{i=1}^{N} (d_i - d_m)^2 \quad (14)$$

- The standard deviation, which represents the mean of all the elementary distances around the mean distance:

$$\sigma = \sqrt{V(v_d)} \quad (15)$$

- The confidence interval around the mean distance can be defined using these parameters. This interval includes a large percentage of the initial elementary distances. Usually, the most adopted degree of confidence is equal to 95.4%. This degree leads to a confidence interval, limited as follows:

$$[d_m - 2\sigma,\ d_m + 2\sigma] \quad (16)$$

This means that if we consider a new image to be segmented, the distance between both structures of interest belongs to the interval at 95.4%. A compact formulation of the distance between structures can be defined by:

$$d = d_m + 2\varphi\sigma, \quad (17)$$

With φ is a real parameter in the interval $[-1, 1]$. The Equation 17 defines then the statistical distance model. This model represents thus a priori knowledge based on the variation of the distance between structures. It can be effectively used in the localization phase, to constrain the evolution of the initial shapes. For that purpose, we should calculate at each iteration, the parameter φ as a function of the current distance d_c (distance between the two shapes in the current iteration). Defined as follows:

$$\varphi = \frac{d_c - d_m}{2\sigma} \quad (18)$$

There are then three possible cases:

$$\begin{cases} \text{If } \varphi \in [-1,1] \text{ then valid distance} \\ \quad If\ \varphi > 1 \text{ then } \varphi \leftarrow 1 \\ \quad If\ \varphi < -1 \text{ then } \varphi \leftarrow -1 \end{cases} \quad (19)$$

In this way, we can require that the distance between shapes will always be in the authorized interval.

3.1.2 Construction of the Statistical Orientation Model

Likewise, the statistical orientation model is calculated at the same time as the shapes and distance models. This model is based on the angle variation between both structures to be segment. It consists to calculate the angles between both structures from the training images and try to deduce a compact formulation, which describes the allowed angles.

First, we will calculate the centers of gravity of the studied structures B_{i1} (Gx_{i1}, Gy_{i1}) and B_{i2} (Gx_{i2}, Gy_{i2}), as described in the previous section.

Then, to calculate the angle θ between both structures (that is the angle formed by the intersection of the line passing through the two centers of gravity B_{i1} and B_{i2} and the horizontal axis *ox*, Figure 1) we proceed as follows:

$$a = \frac{Gy_{i2} - Gy_{i1}}{Gx_{i2} - Gx_{i1}} = \tan(ox, B_{1i}\, B_{2i}) = \tan\theta, \quad (20)$$

with a is the slope of the line $(B_{i1}\, B_{i2})$

$$\theta = \tan^{-1}(a) \quad (21)$$

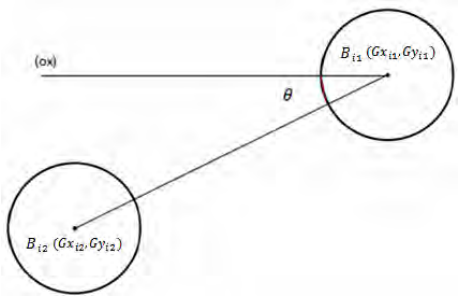


Figure 1: Representation of the angle θ between the reference object (right) and the target object (left)

Similarly, we can calculate the angles between both structures of interest through all the images of the training set. Thereby obtaining an N-dimensional vector angles:

$$v_\theta = (\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N) \quad (22)$$

The aim now is to deduce a compact formulation that describes authorized angles. Indeed, from the vector v_θ , we can calculate the following basic statistical parameters:

- The mean angle between two structures of interest:

$$\theta_m = \frac{1}{N} \sum_{i=1}^{N} \theta_i\ , \quad (23)$$

The variance which measures the dispersion of elementary angles θ_i around the mean angle:

$$V(v_\theta) = \frac{1}{N} \sum_{i=1}^{N} (\theta_i - \theta_m)^2 \quad (24)$$

- The standard deviation, which represents the mean of all the elementary angles around the mean angle:

$$\sigma_1 = \sqrt{V(v_\theta)} \quad (25)$$

The confidence interval around the mean angle can be defined using these parameters. This interval includes a large percentage of the initial elementary angles. Usually, the most adopted degree of confidence is equal to 95.4%. This degree leads to a confidence interval, limited as follows:

$$[\theta_m - 2\sigma_1,\ \theta_m + 2\sigma_1] \quad (26)$$

This means that if we consider a new image to be segmented, the angle between both structures of interest belongs to the interval at 95.4%.

Finally, a compact formulation of the angle between structures can be defined by:

$$\theta = \theta_m + 2\varphi_1\sigma_1, \quad (27)$$

With φ_1 is a real parameter in the interval $[-1, 1]$. The Equation 27 defines then the statistical orientation model. This model represents thus a priori knowledge based on the variation of the angle between structures. It can be effectively used in the localization phase, to constrain the evolution of the initial shapes. For that purpose, we should calculate at each iteration, the parameter φ_1 as a function of the current angle θ_c (angle between both shapes in the current iteration), defined as follows:

$$\varphi_1 = \frac{\theta_c - \theta_m}{2\sigma_1} \quad (28)$$

There are then three possible cases:

$$\begin{cases} \text{If } \varphi_1 \in [-1,1] \text{ then valid angle} \\ If\ \varphi_1 > 1 \text{ then } \varphi_1 \leftarrow 1 \\ If\ \varphi_1 < -1 \text{ then } \varphi_1 \leftarrow -1 \end{cases} \quad (29)$$

In this way, we can require that the angle between shapes will always be in the authorized interval. This allows avoiding the divergence and the collision of shapes during the evolution and increasing the accuracy of results.

Segmentation Guided by Shape, Distance and Orientation Models

The segmentation procedure is sequential. Indeed, the easiest structure to be obtained is segmented first using the standard ASM. The result will then be used as a reference for the segmentation of other structures, based on a priori knowledge of shape, distance and orientation. Thus, the segmentation process can be simulated by the algorithm 1.

Algorithm 1 Segmentation guided by shape, distance and orientation models

$\bar{V}_r$: *mean_shape_reference_structure*
F_r : *Result_localisation_reference _structure*
$\bar{V}_{cibl}$: *mean_shape_target_structure*
F_{cibl_i} : *Result_localisation _target_iteration_i*
F_{cibl_i}' : *Result_ intermediate _iteration_i*
d_c : *current Distance*
θ_c : *current Angle*
%%%%Segmentation to the reference structure
F_r=*procedure_segmentation_ASM* $(\bar{V}_r, V_r = \bar{V}_r + P_r b_r)$
%%%% Segmentation target structure
i=0
While (convergence==no and $i <$ *nbr_max_iterations)*
1. F_{cibl_i}' =*procedure_segmentation_ASM* $(F_{cibl\_i}, V_{cibl} = \bar{V}_{cibl} + P_{cibl} b_{cibl})$
2. d_c = *distance* (F_r, F_{cibl_i}')
3. θ_c = *angle* (F_r, F_{cibl_i}')
4. $(F_{cibl\_(i+1)})$=*limitation_distance_angle*$(d_c, \theta_c, F_r, F'_{cibl\_i}, d = d_m + 2\varphi\sigma, \theta = \theta_m + 2\varphi_1\sigma_1)$
5. *Convergence=compare* $(F_{cibl\_i}, F_{cibl\_(i+1)})$
6. *i=i+1*

End

The limitation by distance and orientation constraint can be simulated by algorithm 2.

Algorithm 2 limitation by distance and orientation constraint

v, w : *real variables*
F_x : *coordinate of the target shape,* F_y : *ordinate of the target shape,* F'_x: *new coordinate of the target shape,* F'_y: *n ordinate of the target shape*
d_m : *mean distance,* σ : *standard deviation_distance,* d_c : *current distance,* φ : *real parameter_distance,* d_{min} : *minimum distance,* d_{max} : *maximum distance*
θ_m : *mean angle,* σ_1 : *standard deviation_angle,* θ_c: *current angle,* φ_1 :*real parameter_angle,* θ_{min} : *minimum angle,* θ_{max} : *maximum angle*

If $\varphi < -1$ *then #* $(d_c < d_{min})$
$v = d_{min} - d_c$
If $\varphi_1 < -1$ *then #* $(\theta_c < \theta_{min})$
$w = \theta_{min} - \theta_c$
$F'_x = F_x cos(w) - F_y sin(w) - v$
$F'_y = F_x sin(w) + F_y cos(w) - v$
If $\varphi_1 > 1$ *then #* $(\theta_c > \theta_{max})$
$w = \theta_c - \theta_{max}$
$F'_x = F_x cos(w) + F_y sin(w) - v$
$F'_y = -F_x sin(w) + F_y cos(w) - v$
Else
$F'_x = F_x - v$
$F'_y = F_y - v$
If $\varphi > 1$ *then #* $(d_c > d_{max})$
$v = d_c - d_{max}$
If $\varphi_1 < -1$ *then #* $(\theta_c < \theta_{min})$
$w = \theta_{min} - \theta_c$
$F'_x = F_x cos(w) - F_y sin(w) + v$
$F'_y = F_x sin(w) + F_y cos(w) + v$
If $\varphi_1 > 1$ *then #* $(\theta_c > \theta_{max})$
$w = \theta_c - \theta_{max}$
$F'_x = F_x cos(w) + F_y sin(w) + v$
$F'_y = -F_x sin(w) + F_y cos(w) + v$
Else
$F'_x = F_x + v$
$F'_y = F_y + v$
Else
If $\varphi_1 < -1$ *then #* $(\theta_c < \theta_{min})$
$w = \theta_{min} - \theta_c$

$$F'_x = F_x\,cos(w) - F_y\,sin(w)$$
$$F'_y = F_x\,sin(w) + F_y\,cos(w)$$
If $\varphi_1 > 1$ *then* $\#\left(\theta_c > \theta_{max}\right)$
$w = \theta_c - \theta_{max}$
$$F'_x = F_x\,cos(w) + F_y\,sin(w)$$
$$F'_y = -F_x\,sin(w) + F_y\,cos(w)$$
Else
$$F'_x = F_x$$
$$F'_y = F_y$$
End
End
End

4. APPLICATION TO STRIATUM SEGMENTATION IN MRI

The striatum is a nervous subcortical structure which consists of the caudate nucleus and putamen. It is a pair structure. This structure is responsible for many functions such as the execution of our movements (voluntary or automatic) and pain management. It is involved in several neurological diseases including Huntington's disease which causes the degeneration of neurons in the striatum in the first place, causing a strongly disturbed motility. In clinical practice, an early diagnosis of Huntington's disease is based, necessarily, on the detection of atrophy of striatum structures. Many segmentation methods have been proposed to contribute to the quantification of striatum atrophy. These models are derived from a statistical learning database [Yan04]. Other works are based on deformable models [Col04]. In [Bab08], the authors present an interesting qualitative and quantitative comparison of the four methods [Alj07, Bab07, Mur07, Pat07] applied for segmentation of internal brain structures on the MRI images, including the caudate nucleus and putamen. The difficulties faced in these applications come mainly from poor definition of these anatomical structures and boundaries. The extraction of these structures is thus often a laborious task.

In this context, we propose a contribution to segment internal brain structures, particularly the caudate nucleus and putamen based on three types of statistical a priori knowledge: the shape of each structure, the distance and the orientation variability between them.

Training step

To model the shapes of the studied structures, we used a training set of 40 brain MRI images (size 256 * 256) from ten different volumes. From each volume, we selected four T1-weighted axial images with the target structures. Then, a labeling step is applied to extract the shapes of both structures: 14 points are used to extract the caudate nucleus and 16 points to extract the putamen.

In the training step, the variability percentage of the original data is fixed at 95% and the length of the grey levels profile is 7 pixels.

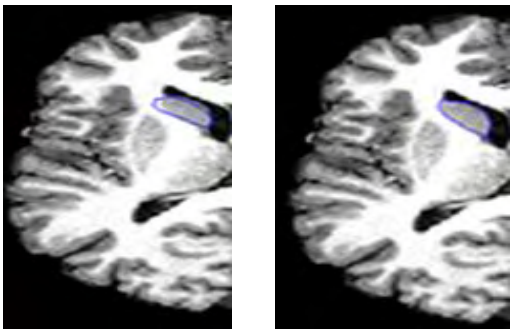
As a result, we ended up building a shape model for each structure (the reference structure is presented by the caudate nucleus and the target structure is presented by the putamen), a distance model and an orientation model, which describes the variation of the distance and the angle between them. The parameters of the obtained models are shown in Table 1.

	Caudate nucleus	**Putamen**
Shapes models	5 principal variation modes	3 principal variation modes
Distance model	Mean distance $d_m = 19.57$ standard deviation_distance $\sigma = 1.66$	
Orientation model	Mean angle $\theta_m = 50.64$ standard deviation_angle $\sigma_1 = 3.34$	

Table 1: Parameters of shapes models, distance model and orientation model

Segmentation Step

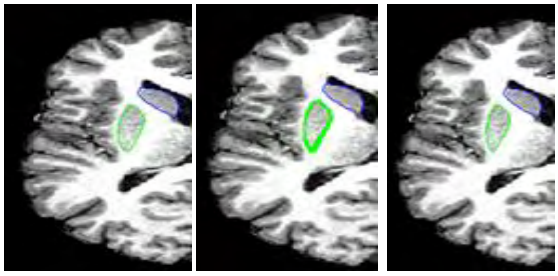
The segmentation procedure is sequential. First, we start with the segmentation of the reference structure based only on the original model (ASM). In this application, after series of tests, we chose the caudate nucleus as a reference structure (the simplest structure to segment). The initialization is the mean shape of the caudate nucleus obtained during training step (figure2.intialization). The segmentation result is illustrated by following figure:



Initialization **Segmentation result**

Figure 2: Segmentation of the reference structure (caudate nucleus) with ASM

Then, we proceed to the putamen segmentation, based on the ASM and spatial relations "ASM+SR". In the various tests, the used initializations are calculated, each time, according to the mean shapes of the putamen obtained during the training step. The maximum number of iterations is set to 60 iterations and the length of the search profile is equal to 21 pixels. In the following, figure 3 shows an example of the segmentation result of the putamen based on ASM+SR, with a good initialization.



(a) (b) (c)

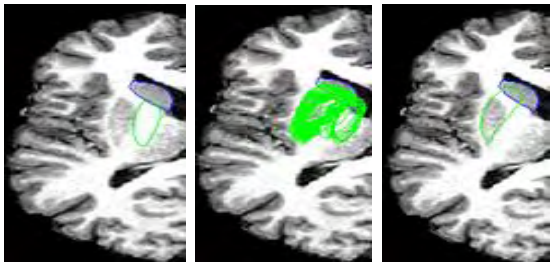
Figure 3: segmentation of the putamen based on ASM+SR. (a) initialization of the mean shape. (b) deformation of the contour. (c) final segmentation result

It is observed that the evolution is performed at a very close neighbor of the target structure. This can provide information on the positive impact of a priori knowledge (shape, distance and orientation) used in the segmentation process.

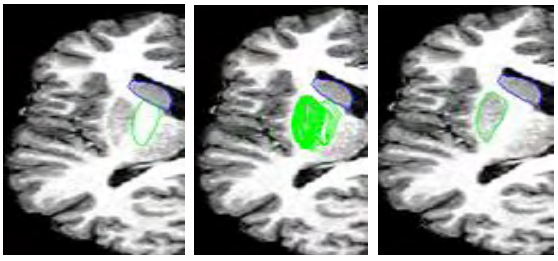
Qualitative evolution

In order to study the behavior of the curve in the evolution process, with and without the constraint of spatial relations, we made a comparison between the proposed method ASM+SR and the original model ASM. The comparison is performed, in each case, on the same image with the same propagation conditions and by adopting different initializations:

- Case 1: close initializations



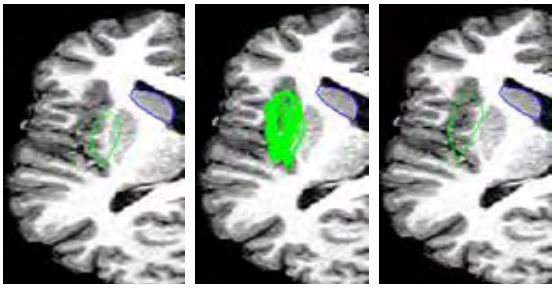
(a)



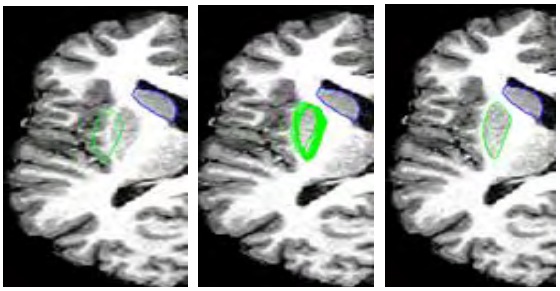
(b)

Figure 4: Examples of obtained results with close initializations. The first column shows the initializations, the second column shows the deformation of the contour and the third column shows corresponding results. (a) Obtained result with ASM. (b) Obtained result with ASM+SR

- Case 2: far initializations



(a)



(b)

Figure 5: Examples of obtained results with far initializations. The first column shows the initializations, the second column shows the deformation of the contour and the third column shows corresponding results. (a) Obtained result with ASM. (b) Obtained result with ASM+SR

Looking at figure 4.a, we can see that if the initialization of the putamen is close to the reference structure, and using original model ASM, the final shape cannot properly define the target structure. There is also a collision between the results. However, in figure 4.b, using the ASM+RS (assuming of course the same initialization), we find that the final shape correctly converged towards the target structure. We can also observe that during evolution, the application of spatial relations make shape gradually pushed towards the target structure. What explains the significant difference between the accuracy of the final result by the ASM+RS and that obtained by ignoring the spatial constraints.

Similarly, by examining figure 5.a and figure 5.b, we see that when ignoring spatial constraints, the final shape diverges to an area where the image intensity is similar to that of Putamen. But the use of spatial constraints helped to push the shape to the target structure and thus obtain a satisfactory result.

In conclusion, these results can provide information on the positive contribution of integrated spatial relationships. Indeed, the application of spatial constraints (distance and orientation) during the evolution has limited the distortion of the initial shape in an authorized zone and thus prevents the divergence to neighboring areas of similar intensity.

However, it must be said that these results can be enhanced to include more examples in the validation process. We must also think about a quantitative evolution in these results

5. CONCLUSION

We have attempted to validate the proposed model "ASM+RS" on a clinical application: segmentation of the caudate nucleus and putamen in MRI cuts of the brain. The obtained results are promising and show good performance of the proposed model. Indeed, the use of an additional constraint of spatial relations (distance and orientation) in the localization step can constrain the development in the regions of interest and achieve satisfactory results. In most of the tests, the proposed model showed its robustness and stability. However, there are limits and a number of perspectives. Indeed, we have not managed to test our model on pathological subjects, thus the precise quantification of the studied pathologies remains incomplete. This is due to the lack of sufficient data in the problem studied. In addition, we treated the case of segmentation of two objects and the proposed method can be easily extended to locate n structures, which will be addressed in future work. Moreover, this method can be improved by adding other spatial constraints to the active shape model e.g., symmetry, which is an important feature of the medical images.

6. REFERENCES

[Alj07] Aljabar, P., Heckemann, R., Hammers, A., Hajnal, J. and Rueckert, D. Classifier selection strategies for label fusion using large atlas databases, MICCAI 2007.

[Bab07] Babalola, K. O., Petrovic, V., Cootes, T. F., Taylor, J. C., Twining, J. C., Williams, T. G. and Mills, A. Automated segmentation of the caudate nuclei using active appearance models. In 3D Segmentation in the clinic: A grand challenge. Workshop Proceedings, MICCAI 2007.

[Bab08] Babalola, K. O., Patenaude, B., Aljabar, P., Schnabel, J., Kennedy, D., Crum, W., Smith, S., Cootes, T. F., Jenkinson, M. and Rueckert, D. Comparison and Evaluation of Segmentation Techniques for Subcortical Structures in Brain MRI. MICCAI, 2008.

[Bar15] Barhoumi, W., Khlifa, N. and Abidi, M. Integration of a Fuzzy Spatial Constraint into Active Shape Models for ROI Detection in Medical Images Current Medical Imaging Reviews. Vol. 11, No. 1, 2015.

[Bou88] Bourke, P. Calculating the Area and Centroid of a Polygon. July 1988

[Col04] Colliot, O. Représentation, évaluation et utilisation de relations spatiales pour l'interprétation d'images. Application à la reconnaissance de structures anatomiques en imagerie médicale, 2004.

[Coo95] Cootes, T.F., Taylor, C.J., Cooper, D.H., and Graham, J. Active Shape Models Their Training and Application. Computer Vision and Image Understanding, vol.6, pp.38-59, 1995.

[Ett14] Ettaieb, S., Hamrouni, K. and Ruan, S. Statistical models of shape and spatial relation-application to hippocampus segmentation. 9th International Conference on Computer Vision Theory and Applications, Lisbon-Portugal, 2014.

[Fou10] Fouquier, G. Doctorat, Ecole Nationale Supérieure des Télécommunications, 2010.

[Gér00] Géraud, T.H. Reconnaissance de structures cérébrales à l'aide d'un atlas et par fusion d'informations structurelles floues, 12 ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2000). Vol.1, pp.287-295, Paris- France, Février 2000.

[Ham98] Hamarneh, G. Active Shape Models - Part I: Modelling Shape and Gray Level Variations. Proceedings of the Swedish Symposium on Image Analysis, 1998.

[Hud08] Hudelot, C., Atif, J. and Bloch, I. FSRO : une ontologie de relations spatiales floues pour l'interprétation d'images, Revue des Nouvelles Technologies de l'Information-les Relations Spatiales : de la Modélisation à la mise en œuvre. RNTI-E-14, pp.53-84, 2008.

[Jaa11] Jaafar, B. and Khlifa, N. Conception of a 2D active-shape model integrating a spatial relation card based on a fuzzy logic. Proceedings of International Conference on Communications, Computing and Control Applications. Hammamet, Tunisia, 2011.

[Kas87] Kass, M., Witkin, A. and Terzopoulos, D. Snakes: Active contour models. International Journal of Computer Vision, 1(4) 321–331, 1987.

[Mur07] Murgasova, M., Dyet, L., Edwards, A. D., Rutherford, M., Hajnal, J. and Rueckert, D. Segmentation of brain MRI in young children. Acad. Rad, 2007.

[Nem09] Nempont, O. Modèles structurels flous et propagation de contraintes pour la segmentation et la reconnaissance d'objets dans les images. Application aux structures normales et pathologiques du cerveau en IRM, Mars 2009.

[Pat07] Patenaude, B., Smith, S., Kennedy, D. and Jenkinson, M. Bayesian shape and appearance models. Technical report TR07BP1, FMRIB Centre - University of Oxford, 2007.

[Per02] Perchant, A. and Bloch, I. Fuzzy Morphisms between Graphs. Fuzzy Sets and Systems. vol.128, pp.149–168, 2002.

[Yan04] Yang, J., Staib, L. H. and Duncan, J. S. Neighbor-Constrained Segmentation With Level Set Based 3-D Deformable Models. IEEE TMI, vol.23, pp.940-948, 2004.

Handwritten Digit Recognition by Support Vector Machine Optimized by Bat Algorithm

Eva Tuba
Faculty of Mathematics,
University of Belgrade
Studentski trg 16,
11000 Belgrade, Serbia
etuba@acm.org

Milan Tuba
Faculty of Computer Sci.,
John Naisbitt University
Bulevar umetnosti 29,
11070 Belgrade, Serbia
tuba@ieee.org

Dana Simian
Faculty of Science
Lucian Blaga University,
Ion Ratiu Street 5-7,
550012, Sibiu, Romania
dana.simian@ulbsibiu.ro

ABSTRACT

Handwritten digit recognition is an important but very hard practical problem. This is a classification problem for which support vector machines are very successfully used. Determining optimal support vector machine is another hard optimization problem that involves tuning of the soft margin and kernel function parameters. For this optimization we adjusted recent swarm intelligence bat algorithm. We intentionally used weak set of features, four histogram projections, to prove that even under unfavorable conditions our algorithm would achieve acceptable results. We tested our approach on standard MNIST benchmark datasets and compared the results with other recent approaches from literature where our proposed algorithm achieved better results i.e. higher correct classification percentage.

Keywords

Handwritten digit recognition, swarm intelligence, bat algorithm, support vector machine, parameter tuning.

1 INTRODUCTION

Nowadays many different applications need some object recognition and because of that it represents an active research field. Object recognition is a part of computer vision, which refers to the problem of recognition of specific object in digital image or digital video. Optical character recognition (OCR) is one subfield of object recognition while digit recognition is the widely studied part of OCR. Digit recognition is used in post offices for sorting the mail [NS12], in banks for reading checks [MAK10], for license plate recognition [CGA08], street number recognition [SCL12], etc.

Task of digit recognition can be divided into two groups, printed digit recognition and handwritten digit recognition. Recognition of printed digits is easier compared to the handwritten digit recognition because printed digits have regular shape and difference between images of the same number are just in the angle of view, size, color, etc. On the other hand, there are numerous handwriting styles which mean that the same digit can be written in many different ways, hence more effort is required to find similarity between instances of the same digit.

One of the oldest techniques for object recognition is template matching. This technique is not suitable for handwritten digit recognition due to numerous variants in writing style, angle of writing, etc. In general, nowadays digit recognition contains three parts, preprocessing, feature extraction and classification. Preprocessing prepares image for feature extraction. Some of the common preprocessing steps are binarization, centering, morphological operations and more. Feature extraction is very important step and success of the classification strongly depends on it. Many different features were proposed in literature. In [JSDK13] horizontal and vertical projection with dynamic thresholding was proposed. Projection histograms are usually used for printed digit recognition and combined with other feature sets. Invariant moments such as geometric moments, affine invariant moments, Legendre moments, Zernike moments, Hu moments, etc. are the common choices for features [SSN16].

One of the most important parts of object recognition algorithms and handwritten digit recognition algorithms is classification. Classification in computer science represents prediction of class or label for an object based on its similarity with previous objects. In machine learning, each object or instance is represented with same set of features. Based on the learning

algorithm, classifiers can be divided to unsupervised and supervised classifiers. Supervised learning uses knowledge of labels for instances used for building the model while instances for unsupervised learning are unlabeled. Today, many techniques for building a classification model are used. One of the simplest machine learning algorithms is k-nearest neighbors (KNN). KNN is nonparametric technique that classifies instances by a majority vote of its neighbors. Instances will be assigned to the class most common amongst its K nearest neighbors measured by a distance function such as Euclidean distance, Manhattan distance, Hamming distance, etc. Decision tree represents rule based classifier widely used in different applications [ZWPJ14]. Some of the classifiers work with probability model and use statistical learning algorithms. These classifiers calculate probability that an instance belongs to each class. Linear combination of features that best describes difference between classes was found with these classifiers [CLY+11]. One of the recent proposed and widely used perception based classifier is artificial neural network (ANN). ANN is also used for prediction and pattern recognition [MLW+13b], [KPB08]. In the past few years one of the most used and most successful classifiers is support vector machine (SVM) [CV95]. Many applications use SVM for solving the classification problem, especially these for handwritten digit recognition. In [MUS08], SVM was used to improve classification accuracy for the OCR of mathematical documents. In [LMPS+15] it was used for classification of brain metastasis and radiation necrosis. In [GC04] support vector machines and neural network were combined for classification of handwritten digit recognition.

In this paper we propose using SVM for handwritten digits recognition. Support vector machine has a few parameters that should be adjusted. First parameter is parameter of soft margin C that allows outliers to be misclassified. In real life data, outliers are common and also data usually are not linearly separable. In that case some kernel functions need to be used and parameter of this functions also need to be tuned. One of the common kernel functions is Gaussian radial basis function with parameter γ. Tuning parameters of SVM is a hard optimization problem.

Bio-inspired algorithms such as swarm intelligence algorithms are widely researched and used for hard optimization problems. In swarm intelligence algorithms behavior of collectives of simple agents were simulated. Particle swarm optimization (PSO) is one of the oldest algorithm in this class of algorithms [KE95]. Today many different algorithms were proposed and used such as ant colony algorithm, artificial bee colony, cuckoo search, firefly algorithm and others.

Swarm intelligence algorithms have been used for SVM parameters tuning. In [BHX13] memetic algorithm based on PSO and pattern search was used for SVM parameters tuning. Modified PSO that uses chaotic mapping was introduced in [Wu11] for parameter optimization of SVM variant called wavelet v-support vector machine and in [LZ15] for improving classification accuracy of linear square SVM. In [MYK14] artificial bee colony was used for parameter tuning for linear square SVM. In [XBH14] firefly algorithm was used for optimization of multi-output support vector machine. A parallel time variant particle swarm optimization algorithm to simultaneously perform the parameter optimization and feature selection for SVM was proposed in [CYW+14].

In this paper we propose using SVM optimized by bat algorithm for handwritten digit recognition using intentionally weak features with which other approaches would not give good results. We propose usage of recent swarm intelligence algorithm, bat algorithm, for SVM parameter tuning. Our proposed algorithm was tested on standard MNIST [LBBH98] dataset for handwritten digit recognition and performance was better than other approaches from literature [MLW+13b], [KDB+13].

The rest of the paper is organized as follows. In Section 2 mathematical model for SVM is described. Section 3 then describes the bat algorithm. Explanation of our proposed algorithm for SVM parameter tuning is given in Section 4. Experimental results and comparison with other approaches from literature are given in Section 5. At the end conclusion along with proposed future work are presented in Section 6.

2 SUPPORT VECTOR MACHINE

Support vector machine was proposed by Vapnik as binary classifier [CV95]. It represents one of the latest supervised learning classifiers and it was used in numerous applications. SVM discovers a hyperplane that separates data from different classes. Each instance is labeled with one of existing classes and they are represented as points in space. SVM builds a model based on instances from training set and further classification of unknown instances is done by that model.

Hyperplane that separates labeled instances from the training set is defined by the next equation:

$$y_i(w \cdot x_i + b) \geq 1 \quad \textbf{for} \quad 1 \leq i \leq n. \tag{1}$$

where $x_i \in R^d$ are instances represented as vectors in d-dimensional space, n is the number of instances, $y_i \in \{-1, 1\}$ are classes of corresponding instances and w and b are parameters of the hyperplane. This hyperplane is determined by the nearest instances that

are called support vectors. Hyperplane should be as far as possible from instances of both classes. The distance that should be maximized is $\frac{2}{||w||}$.

The described model has a problem to classify real life data, because all instances must be on the correct side of the hyperplane. Real world data contains some noise and usually a few outliers. The previous model is not able to separate such data. As a solution for this problem, using of soft margin was proposed. Soft margin is used instead of Eq. (1). The idea is to introduce a slack variable ε that allows some instances to be misclassified i.e. to be on the wrong side of the hyperplane. This is defined by the following expression:

$$y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \quad 1 \leq i \leq n \qquad (2)$$

Finding this hyperplane is done by solving the following quadric programming problem:

$$\min \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\varepsilon_i, \qquad (3)$$

where C is the soft margin parameter. Increasing the value of parameter C asymptotically leads to the model with hard margin. Selecting appropriate value for this parameter has major influence on classification accuracy [Wan05].

Another problem with this model, when it comes to real world data, is the assumption that instances are linearly separable. In order to make SVM suitable for non-linearly separable data kernel function is used instead of dot product. Theoretically, any function that satisfies Mercer's condition can be used as kernel function. In practice, usually Gaussian radial basis function (RBF), polynomial function and sigmoid function are used. Kernel function projects data into higher dimensional space in order to make it linearly separable. In this paper we used RBF as kernel function. RBF is defined by the next equation:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2). \qquad (4)$$

where γ is the parameter of kernel function. This parameter has influence on the quality of classifier, so tuning the value of it is an important task. Too large value of γ will reduce benefits gained by introducing the kernel function and too small value will make decision boundary sensitive to the noise in training data.

Selecting optimal values for SVM's parameters is very important task. In [HCL10] most common technique for parameters tuning, grid search with cross validation, was described. Grid search builds models for different values of parameters and checks the accuracy of these models. Cross validation is used for determination of the model's accuracy. Training set is divided into v distinct subsets. For training $v-1$ subsets were used and the accuracy was checked on the remaining subset. All subsets are used as test set once and the accuracy is the average value of v obtained accuracies. This method requires huge computational time and the search for optimal pair of values for (C, γ) is limited to predefined set of values.

Instead of the grid search, different stochastic optimization algorithms were successfully used [BHX13], [Wu11], [CYW+14], [MYK14].

3 BAT ALGORITHM

Bat algorithm is one of the recent swarm intelligence algorithm introduced by Yang [Yan10]. The algorithm was inspired with echolocation of bats which they use to detect pray and avoid obstacles. Bats emit sound pulses and navigate by using the time delay from emission to reflection.

Bat algorithm was widely used and studied in the past few years. In [YG12] it is used for multi-objective optimization and in [GYAT12] constrained optimization. In [HZL13] global engineering optimization and large-scale optimization problems were solved by bat algorithm. As a result of wide research of algorithm many improvements and hybridizations were developed [AT14], [FJFY13]. Numerous applications use bat algorithm for some real world hard optimization problems such as image processing [ZW12], RFID network planing [TB15], training neural networks [TAB15], etc.

Bat algorithm starts with initialization of the population that is performed randomly. Each bat from the population is represented by its location x_i^t, velocity v_i^t, frequency f_i^t, loudness A_i^t and the emission pulse rate r_i^t in a D-dimensional search space. Location and velocity are updated at each iteration based on the previous solution. The new solution is calculated according to the following equations [Yan10]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad (5)$$

$$v_i^t = v_i^{t-1} + (x_* - x_i^{t-1})f_i \qquad (6)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (7)$$

where β is a random vector generated from uniform distribution from the closed range [0, 1] and x_* represents the current global best location which is found after comparing all the solutions among all the bats. At the beginning each bat is randomly assigned a frequency which is drawn uniformly from the interval $[f_{min}, f_{max}]$.

Every swarm intelligence algorithm has two important operations, exploration and exploitation. In the bat

algorithm for local search (exploitation) random walk with direct exploitation is used. It is defined by the following equation:

$$x_{new} = x_* + \varepsilon * A_t \tag{8}$$

where $A_t = < A_i^t >$ represents the average loudness of all bats at the time step t and ε is random number from the range $[-1,1]$. Parameter ε defines intensity and direction of random walk. The local search depends on the rate r_i of pulse emission for the i-th bat. When bat approaches to its pray, bat becomes more silent, so the loudness decreases, but the pulse rate increases. For the purpose of the algorithm, the pulse rate can be defined in the range from 0 to 1, where 0 means that there is no emission at all and 1 means that the bat is emitting at their maximum [Yan10]. It can be formally represented by the next equations:

$$A_i^t = \alpha A_i^{t-1} \tag{9}$$

$$r_i^t = r_i^0(1 - e^{-\gamma t}) \tag{10}$$

where α and γ represent constants defined according to the problem that is solved by the bat algorithm.

Bat algorithm is summarized in Algorithm 1.

Algorithm 1 Pseudo-code for the original bat algorithm [Yan10]

```
Define the objective function f(x), x = (x_1, x_2, ..., x_d)^T
Initialize the population of bats x_i, (i = 1, 2, ...n) i v_i
Define pulse frequency f_i at the position x_i
Initialize pulse rates emission r_i and sound loudness A_i
while t < IN do
    Generate new solutions by adjusting frequency and updating
    velocities and locations
    solutions
    by using equations (5) - (7)
    if rand > r_i then
        Select the best solution from the population
        Generate new solution in the neighborhood of chosen
        solution
    end if
    Generate new solution by flying randomly (random walk)
    if rand < A_i and f(x_i) < f(x_*) then
        Accept new solutions
        Increase r_i and decrease A_i
    end if
    Rank all the bats in the population and find the current best
    solution x_*
end while
Post-process results and visualization
```

4 OUR PROPOSED ALGORITHM

In this paper we proposed using projection histograms as the feature set for handwritten digits. Projection histograms were usually used for typed digit recognition (e.g. license plate recognition). For handwritten digits recognition projection histograms were not much used, especially without another set of features. We intentionally used this weak set of features to test our SVM classifier under unfavorable conditions. Fig. 1 shows example of projection histograms on x-axis for all 10 digits.

Figure 1: Projection histogram on x-axis ($y = 0$)

Because of various writing styles, thickness of pen, angle, etc., projection histograms can be very different for the same digit and on the other hand they can be very similar for different digits so projection on one axis cannot be sufficient. Fig. 2 shows example of histograms for digits 0, 3 and 8.

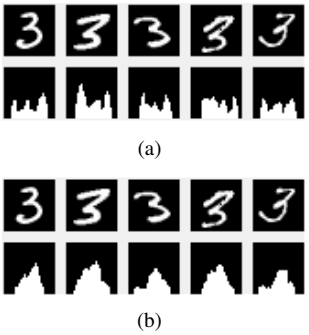
(a) (b)

Figure 2: Example of histograms for numbers 0, 8 and 3 on (a) x and (b) y axis

It can be noticed that projection histograms on x axis for digits 8 and 3 are very similar, they have peak in the middle, but projections on y axis are different. Number 3 has three peaks, while number 8 has little dent in the middle. On the other hand projection histograms for digits 8 and 0 on y axis are similar, but difference is clear at projection histograms on x axis.

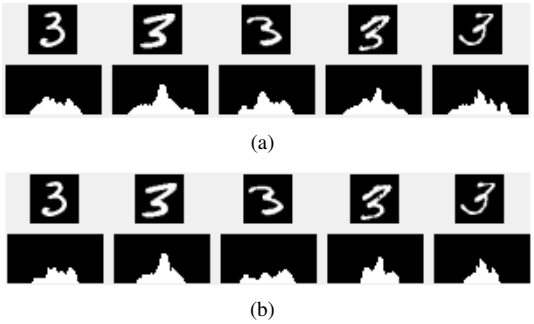
Besides these two projection histograms, in our algorithm we used two more projections, on lines $y = x$ and $y = -x$, thus each digit was represented with four different projection histograms. Fig. 3 and Fig. 4 show examples of all four histograms for different samples of digit 3. It can be seen that projection histograms on one axis can be very different, but with the same characteristics, and combination of four histograms helps to differentiate between different digits.

Described feature set was used as input for support vector machine. For handwritten digit recognition, ten different classes are needed, one class for each digit. SVM is binary classifier while for this task multi-classification is needed. Two main techniques are used in cases like this. First, known as *one-against-all*, makes one model for each class. Each model separates one class from all others. This method is more suitable for classifiers that produce real valued probabilities that instance belongs to class. Second



(a) (b)

Figure 3: Number 3 histograms on (a) x and (b) y axis



(a) (b)

Figure 4: Number 3 histograms on (a) $y = x$ and (b) $y = -x$

method that is used for multi-classification with binary classifiers is *one-against-one*. If there are n classes then $\frac{n(n-1)}{2}$ models need to be made, one model for each pair of classes. Class of an unknown instance can be determined by counting the votes. Each model produces result and class that was determined most times represents the class of unknown instance. In the case when two or more classes have the same number of votes, different methods can be used for making the final decision.

In this paper we proposed combination of the two mentioned techniques for multi-classification. Initially, classes were predicted by 10 different models (*one-against-all*). If the class was not determined uniquely or was not determined at all, we used *one-against-one* technique.

Important part of classification procedure is scaling. Feature values of training and test data should be scaled to range $[0,1]$ or $[-1,1]$. Scaling values have significant influence on classification accuracy. Without scaling data in greater numerical range would dominate over data in smaller range. Also training and test data should be scaled with same factor.

Parameters of the SVM were tuned by bat algorithm. Dimension of search space was 2, search for optimal pair of values for C and γ. Objective function was to maximize accuracy of the SVM models. Accuracy was calculated with 10-fold cross validation as it was described in Section 2.

For different problems, parameters of the bat algorithm should be adjusted. Besides parameter adjustment, some other modifications may be needed. Pulse rate r and loudness A can be static for each bat or they can be changed according to Eq. 9 and Eq. 10. Speed of convergence of these two parameters is determined by the values of α and γ. If pulse rate increases too fast, probability of random walk will be low. In order to ensure random walk, initial pulse rate should be closer to 1, so random walk would be performed in at least $1 - r^0$ fraction of cases. Random walk will be performed even in the later cycles of the algorithm. Low values of loudness provide exploration. If loudness increases too fast it is possible to be trapped in local optima. Based on loudness, solution can be accepted even if it is not better than the current solution. This provides exploration and decreases the possibility of being trapped in local optimum. Loudness will increase with number of iteration, thus in later iterations in less cases generated solution would be accepted if it is not better. Another important parameter is frequency. Based on the range for frequency, new solution will be generated in some space around the global best position. For frequency value 1, new solution will be generated at the same point as the best solution. Large range for frequency allows wider space around the best solution for new solution. Depending on problem different frequency ranges should be used.

5 EXPERIMENTAL RESULTS

Quality of our proposed algorithm for handwritten digit recognition was tested on standard MNIST database [LBBH98]. In this database images were preprocessed so in this paper preprocessing was not included. All images were centered in a 28×28 image. This database contains 60,000 images for training and 10,000 images for testing. We tested our algorithm on limited set of digits. Fig. 5 shows example of images from MNIST database.

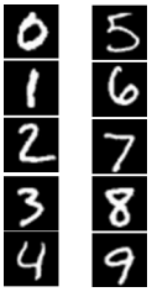


Figure 5: Example of digits from MNIST dataset

Proposed algorithm was implemented in Matlab R2015a and for classification LIBSVM (Version 3.21) [CL11] was used. Experiments were performed on the platform with the following features: Intel

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	99	0	0	0	0	0	0	0	1	0
1	0	99	0	0	0	0	0	0	1	0
2	0	0	97	1	1	0	0	0	1	0
3	0	0	3	89	0	1	0	1	6	0
4	0	0	0	0	98	1	0	0	0	1
5	0	0	1	7	0	91	0	0	1	0
6	0	0	0	0	0	0	100	0	0	0
7	0	0	0	0	0	1	0	93	0	6
8	0	0	1	0	0	1	2	1	95	0
9	1	0	0	0	2	0	0	1	1	95

Table 1: Accuracy of classification for our proposed method (%)

® Core^{TM}i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS.

Parameters for bat algorithm were determined empirically, using theoretical insight from Section 4. Initial value of pulse rate was set to γ=0.92 and loudness was α=0.993. Frequency range was $[-8,10]$.

Dimension of images are 28 by 28, so projection histograms on x and y axis contain 28 elements. Projection histograms on $y = x$ and $y = x$ have 55 elements. Using combination of all four histograms as input vectors means that each instance contains 166 elements.

Search space for exponents of parameters of SVM were $[0,20]$ for C and for γ was $[-15,5]$ so the search space for parameters were $[2^0,2^{20}]$ and $[2^{-15},2^5]$ since log scaling was used. Computational times were few minutes for each model but they are not very important since the model is constructed only once and offline.

Table 1 shows the results of classification. The best accuracy was achieved for digit 6 where all samples were recognized correctly. Drastically worse accuracy compared to other digits was achieved for recognition of digit 3. Digit 3 was recognized as 8 in 6% of cases and it was classified as 2 in 3% of cases. Only 89% of samples of digit 3 were recognized correctly.

Additionally, we compared our results with other state-of-the-art algorithms. In [KDB+13] use of multilayer neural network (MLNN) was proposed and dilatation algorithm combined with zoning techniques was used. Table 2 shows comparison of results reported in [KDB+13] and the results obtained with our proposed method.

In [KDB+13] the worst results were achieved for recognition of digit 9, while the best result was for digit 1. Our proposed model achieved significantly better results for each digit, thus the global accuracy was better as well. Our proposed feature set does not require any zoning technique so it is simpler to extract the features compared to [KDB+13] and the accuracy of classification is better.

Digit	**MLNN**	**SVM-BAT**
0	86.45	99.00
1	94.39	99.00
2	88.73	97.00
3	77.02	89.00
4	76.12	98.00
5	84.10	91.00
6	78.81	100.00
7	77.12	93.00
8	79.03	95.00
9	49.64	95.00
Global	79.14	**95.60**

Table 2: Accuracy of classification reported in [KDB+13] and our proposed method (%)

Another recent algorithm from literature was [MLW+13b] where three machine learning algorithms were proposed for handwritten digit recognition, extreme learning machine (ELM), regularized extreme learning machine (REML) and optimal weight learning machine (OWLM). Neural networks with different number of nodes were tested and compared. The best results were achieved by OWLM with 150 nodes. Man et al. in [MLW+13b] reported 85.16% as the best global accuracy, which is significantly less than accuracy of 95.60% achieved with our proposed method. With ELM learning algorithm and 150 nodes accuracy was 82.83% and with REML the highest accuracy that was achieved was 82.96%. Our proposed method produced better results compared to all results presented in [MLW+13b].

6 CONCLUSION

In this paper we proposed a novel algorithm for handwritten digit recognition. The goal was to use simple feature set as input for support vector machine that was used for classification. Optimal SVM models were determined by recent swarm intelligence algorithm, bat algorithm. Bat algorithm was adjusted and used for parameter tuning of the support vector machine. We tested our proposed method on standard

MNIST dataset and achieved global accuracy of 95.60%. We compared our method with other methods proposed in literature [KDB+13], [MLW+13b] and our proposed method obtained better accuracy with rather simple feature set. This establishes this approach as very robust and by using more complex features the results could be further improved. Additional validation can be done using other databases, for example USPS.

7 ACKNOWLEDGMENT

M. Tuba was supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

D. Simian was supported by the research grant LBUS-IRG-2015-01, project financed by Lucian Blaga University of Sibiu.

8 REFERENCES

[AT14] Alihodzic, A. and Tuba, M. Improved hybridized bat algorithm for global numerical optimization. The 16th IEEE International Conference on Computer Modelling and Simulation, UKSim-AMSS 2014, pp. 57–62, 2014.

[BGC12] Bhattacharya, G., Ghosh, K. and Chowdhury, A. S. An affinity-based new local distance function and similarity measure for kNN algorithm. Pattern Recognition Letters, 33(3):356–363, 2012.

[BHX13] Bao, Y., Hu, Z., and Xiong, T. A PSO and pattern search based memetic algorithm for SVMs parameters optimization. Neurocomputing, 117:98–106, 2013.

[CGA08] Caner, H., Gecim, H. S. and Alkar, A. Z. Efficient embedded neural-network-based license plate recognition system. IEEE Transactions on Vehicular Technology, 57(5):2675–2683, 2008.

[CL11] Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3):27:1–27:27, May 2011.

[CLY+11] Chen, H.-L., Liu, D.-Y., Yang, B., Liu, J. and Wang, G. A new hybrid method based on local Fisher discriminant analysis and support vector machines for hepatitis disease diagnosis. Expert Systems with Applications, 38(9):11796–11803, 2011.

[CV95] Cortes, C. and Vapnik, V. Support-vector networks. Machine Learning, 20(3):273–297, 1995.

[CYW+14] Chen, H.-J., Yang B., Wang, S.-J., Wang G., Liu, D.-Y., Li, H., and Liu, W.-B. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy. Applied Mathematics and Computation, 239:180–197, 2014.

[FJFY13] Fister, I. Jr., Fister, D., and Yang, X.-S. A hybrid bat algorithm. Elektrotehniski Vestnik/Electrotechnical Review, 80(1-2):1–7, 2013.

[GC04] Gorgevik, D. and Cakmakov, D. An efficient three-stage classifier for handwritten digit recognition. The 17th International Conference on Pattern Recognition (ICPR 2004), Volume 4, pp. 507–510, 2004.

[GYAT12] Gandomi, A. H., Yang, X.-S., Alavi, A. H. and Talatahari, S. Bat algorithm for constrained optimization tasks. Neural Computing and Applications, 22(6):1239–1255, 2012.

[HCL10] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. A practical guide to support vector classification. Technical report, National Taiwan University, 2010.

[HZL13] Huang, G.-Q., Zhao, W.-J. and Lu, Q.-Q. Bat algorithm with global convergence for solving large-scale optimization problem. Application Research of Computers, 30(5):1323–1328, 2013.

[JSDK13] Jagannathan, J., Sherajdheen, A., Deepak, R. M. V. and Krishnan, N. License plate character segmentation using horizontal and vertical projection with dynamic thresholding. International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), pp. 700–705, 2013.

[KDB+13] Kessab, B. E., Daoui, C., Bouikhalene, B. Fakir, M. and Moro, K. Extraction method of handwritten digit recognition tested on the MNIST database. International Journal of Advanced Science and Technology, 50(6):99–110, 2013.

[KE95] Kennedy, J. and Eberhart, R. Particle swarm optimization. IEEE International Conference on Neural Networks, 1995., Volume 4, pp. 1942–1948, 1995.

[KPB08] Kang, M. and Palmer-Brown, D. A modal learning adaptive function neural network applied to handwritten digit recognition. Information Sciences, 178(20):3802 – 3812, 2008.

[LBBH98] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp. 2278–2324, 1998.

[LMB+11] Lee, C.-C., Mower, E., Busso, C. Lee, S. and Narayanan, S. Emotion recognition using a hierarchical binary decision tree approach. Speech Communication, 53(9-10):1162–1171, 2011.

[LMPS+15] Larroza, A., Moratal, D., Paredes-Sanchez, A., Soria-Olivas, E., Chust, M. L., Arribas, L. A. and Arana, E. Support vector machine classification of brain metastasis and radiation necrosis based on texture analysis in MRI. Journal of Magnetic Resonance Imaging, 42(5):1362–1368, 2015.

[LZ15] Liu, F. and Zhou, Z. A new data classification method based on chaotic particle swarm optimization and least square-support vector machine. Chemometrics and Intelligent Laboratory Systems, 147(October):147–156, 2015.

[MAK10] Mahmoud, S. A. and Al-Khatib, W. G. Recognition of Arabic (Indian) bank check digits using log-Gabor filters. Applied Intelligence, 35(3):445–456, 2010.

[MLW+13b] Man, Z., Lee, K., Wang, D., Cao, Z. and Khoo, S. An optimal weight learning machine for handwritten digit image recognition. Signal Processing, 93(6):1624–1638, 2013.

[MUS08] Malon, C., Uchida, S. and Suzuki, M. Mathematical symbol recognition with support vector machines. Pattern Recognition Letters, 29(9):1326–1332, 2008.

[MYK14] Mustaffa, Z., Yusof, Y. and Kamaruddin S. S. Enhanced artificial bee colony for training least squares support vector machines in commodity price forecasting. Journal of Computational Science, 5(2):196–205, 2014.

[NS12] Niu, X.-X. and Suen, C.-Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognition, 45(4):1318–1325, 2012.

[SCL12] Sermanet, P., Chintala, S. and LeCun, Y. Convolutional neural networks applied to house numbers digit classification. The 21st International Conference on Pattern Recognition (ICPR), pp. 3288–3291, 2012.

[SSN16] Singh, P.-K., Sarkar, R. and Nasipuri, M. A study of moment based features on handwritten digit recognition. Applied Computational Intelligence and Soft Computing, 2016:1–17, 2016.

[TAB15] Tuba, M., Alihodzic, A. and Bacanin, N. Cuckoo search and bat algorithm applied to training feed-forward neural networks. In: Recent Advances in Swarm Intelligence and Evolutionary Computation, volume 585 of Studies in Computational Intelligence (Editor:Xin-She Yang), pp. 139–162. Springer International Publishing, 2015.

[TB15] Tuba, M. and Bacanin, N. Hybridized bat algorithm for multi-objective radio frequency identification (RFID) network planning. IEEE Congress on Evolutionary Computation (CEC2015), pp. 499–506, 2015.

[Wan05] Wang, L. Support Vector Machines: Theory and Applications. Series: Studies in Fuzziness and Soft Computing, Volume 177, Springer-Verlag Berlin Heidelberg, 2005.

[Wu11] Wu, Q. A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM. Expert Systems with Applications, 38(1):184–192, 2011.

[XBH14] Xiong, T., Bao, Y. and Hu, Z. Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. Knowledge-Based Systems, 55(January):87–100, 2014.

[Yan10] Yang, X.-S. A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Volume 284 of the series Studies in Computational Intelligence: pp. 65–74, 2010.

[YG12] Yang, X.-S. and Gandomi, A. H. Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5):464–483, 2012.

[ZW12] Zhang, J.-W. and Wang, G. Image matching using a bat algorithm with mutation. Applied Mechanics and Materials, 203(1):88–93, 2012.

[ZWPJ14] Zhang, Y., Wang, S., Phillips, P. and Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowledge-Based Systems, 64(July):22–31, 2014.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Face Tracking using a Combination of Colour and Pattern Matching Based on Particle Filter

Matilde Gonzalez
Université Paul Sabatier
118 Route de Narbonne
F-31062 TOULOUSE
CEDEX 9
gonzalez@irit.fr

Christophe Collet
Université Paul Sabatier
118 Route de Narbonne
F-31062 TOULOUSE
CEDEX 9
collet@irit.fr

ABSTRACT

Abstract. Robust real-time face tracking is an important and challenging task in computer vision applications. In this paper, we propose a novel particle filter algorithm to robustly track faces. Particle observations are computed by considering cue and appearance feature. Cue feature is used to identify skin regions, e.i. face and hands, while appearance is used to directly label targets. Normalized Cross Correlation (NCC) between an image template and particle samples is computed to robustly find the face among other skin regions. In other words, the image template is registered to a frame using particle filter to perform the optimization. Real-time is achieved by using integral images to compute image features. Evaluation results show the advantages and limitation of our approach.

Keywords

Face Tracking, Particule filter, Real-time, Robust tracking, Occlusions, Color and appearance particles model

1 INTRODUCTION

Face tracking is a necessary step in many computer vision applications such as gesture recognition, human computer interaction, surveillance systems and sign language analysis [Gia09a][Gre05a][Mit07a]. The presence of noise, occlusions, fast dynamic changes and background complexity make face tracking a hard task. We focus our research in the domain of sign language analysis, and more specificaly sign language corpora automatic annotation. Sign languages are visuo-gestural languages used by deaf community as natural mean of communication. They are studied by linguists and computer scientists mainly through videos corpora of persons (signers) in spontaneous expressions or dialogs. One activity of these research consists in annotating the videos with relevant informations like basically, the glosses associated to signs (words in spoken language associated to the meaning of each sign). In order to facilitate and speed up this task we propose computer vision algorithm to automaticaly track pertinent component of the signer, e.g. hands and head. In the present paper we focus on head tracking which is quite straightforward knowing that there is only one person to track in the video. But sign language expression leads the hands to move very fast, in a rather erratic way and very often to occlude the face. Head tracking makes it easier to track hands in the same time and it allows to segment the hand region of that of the head [Gon11a]. In this context, we don't necessarily care about the head pose or the facial features tracking. The solutions like Active Appearance Model (AAM) with 2D video [Zho10a][Pia10a] or video associated with depth camera [Smo14a], or landmark tracker [Uri15a], are not suitable and they do not handle well occlusions. More complex methods can handle occlusions but needs to be initialised by hand and are resource intensive [Zha13a].

Many tracking algorithms have been proposed to deal with these problems. Deterministic methods are based on a similarity cost function between a template and the current image incorporating, then, *a priori* information [Bra98a][Bir98a][Hag02a]. On the other hand stochastic methods are based on a dynamic model of the system. In the case of linear-Gaussian model, a Kalman filter estimates the posterior probability density function [Ste01a][Jan02a][Kir02a]. For non-Linear or non-Gaussian multi-modal distributions, the particle filter algorithm [Isa98a] has become very popular since it solves the limitation imposed by Kalman Filter.

Particle filter tracking algorithms usually use contours, colour features and appearance models

[Num03a][Mic04a][Gia09a][You10a]. Colour based algorithms have the inconvenient that same model could be used to represent different objects, e.g. skin blobs represent head and hands. Other solutions propose the fusion of several cues [Rad06a][Zha07a]. Particle observations are computed using a linear combination of various features, e.g. colour geometrical features. However it is becomes dependent on the coefficients used in the linear combination.

In this paper, we propose to combine *a priori* information with a dynamic model of the system. The proposed method includes global appearance information, an image template, and colour feature while performing particle filtering. The contributions of our work lie mainly in the fact that (i) multiple features are directly integrated on the computation of particle weights instead of a linear combination of the observation likelihood, (ii) geometric information is implicitly considered by the proposed model while computing colour cue likelihood and (iii) the registration of the template model represents an approach on pattern matching that can be described as image registration with particle filter optimization.

The remainder of this paper is organized as follows. Section 2 describes particle filter principle. Section 3 details the proposed model to combine colour and shape. Section 4 presents the proposed observation model to take into account multiple features. Section 5 shows the evaluation performed and last Section 6 presents our main conclusions.

2 PARTICLE FILTER

Visual tracking intends to estimates the state of the system that changes over time by using a sequence of noisy measurements. Bayes filter compute the posterior probability density function $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of the current state $\mathbf{x}_t$ conditioned on all observations $\mathbf{z}_{1:t} = \mathbf{z}_1 \dots \mathbf{z}_t$ with $\mathbf{z}_t$ to the observation vector obtained at time t. For a first-order Markov process, i.e. the state $\mathbf{x}_t$ depends only on $\mathbf{x}_{t-1}$, the probability density function $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be obtained in two stages: prediction and update. It is derived as

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t}) = k \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) \quad (1)$$

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1})\, dx_{t-1} \quad (2)$$

where k corresponds to a normalization term independent of $\mathbf{x}_t$. Eq.1 represents the update stage where the posterior probability density is computed using the observation likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ and the temporal prior distribution, $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$, over $\mathbf{x}_t$ given past observations. Eq.2 corresponds to the prediction stage where the prior distribution for $t+1$ is estimated by the convolution of the posterior distribution $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ and the transition probability distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, i.e. the dynamic model of the system.

Particle filter (PF) [Bla98a] presents a good solution framework for tracking stochastic movements. It sequentially estimates, using random sampling to approximate the optimal solution, the states $\mathbf{x}_t$ of the system by implementing a recursive Bayesian filter by Monte Carlo simulations. The posterior probability density $p(\mathbf{x}_t|\mathbf{z}_t)$ of the current state $\mathbf{x}_t$ is approximated by a weighted particle sample set, $s_t^n, \pi_t^n{}_{n=1}^N$. PF maintain multiple hypothesis, i.e. each particle is a hypothetical state of the object, weighted by a discrete sampling probability $\pi_t^n \propto p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^n)$. Particle weights correspond to the observation generated by the hypothetical state and reflects the image feature relevance associated to each particle, see Section 4. The state $\mathbf{x}_t$ is finally estimated using the particle set and the associated weights.

The algorithm consists essentially of the following steps:

input : The particle set at time $t-1$: $\{s_{t-1}^n, \pi_{t-1}^n\}_{n=1}^N$

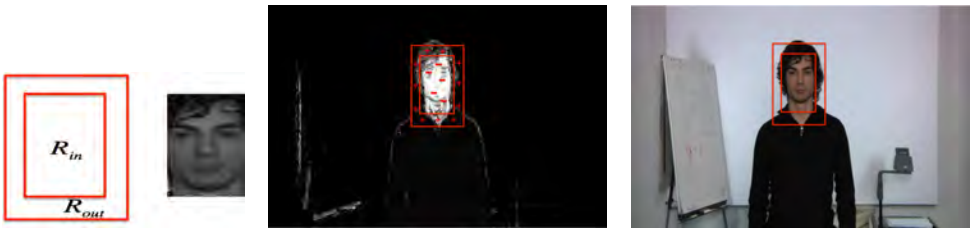
output : The expectation result at time t : $E[\mathbf{x}_t]$

1. **Resample** N particles from the set $\{\mathbf{s}_{t-1}^n, \pi_{t-1}^n\}_{n=1}^N$ to $\{s'^n_t, \frac{1}{N}\}_{n=1}^N$
2. **Propagate** each particle using the dynamic model $\mathbf{s}_t^n \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1} = \mathbf{s}'^n_{t-1})$ to obtain $\{\mathbf{s}_t^n, \frac{1}{N}\}_{n=1}^N$
3. **Weight** particles with the image feature $\mathbf{z}_t$ as $\pi_t^n \propto p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^n)$ and normalize so that $\sum_{n=1}^N \pi_t^n = 1$
4. **Estimate** the tracking result of the object at time t by $E[\mathbf{x}_t] = \sum_{n=1}^N \pi_t^n \mathbf{s}_t^n$

In this work each hypothetical state $\mathbf{s}_t^n = \mathbf{x}_t^n, \mathbf{y}_t^n$ corresponds to a state propagated through a first order auto-regressive process model, $\mathbf{x}_t = \mathbf{x}_{t-1} + \eta$, where η is a zero-mean Gaussian random variable and $\mathbf{x}_t^n, \mathbf{y}_t^n$ are the the coordinates of particle n at time t.

3 OBJECT MODEL

Many methods combine shape and colour information [Ima98a][Mic04a][You10a] to robustly track face and hands. Face is modelled as a rectangular skin blob. These methods handle skin region occlusions by merging and separating blobs. However when hands fully occludes another skin object, one blob may be lost. Other methods use more complex shapes, but they



(a) (b) (c)

Figure 1: (a) Illustrates the proposed face model. (b) shows the rectangular model lying over the skin probability map when ρ_{R_T} is maximal and (c) represents the best matching position for the template registration.

are computationally expensive and time-consuming [Bir98a]. Gianni *et al.* [Gia09a] model each skin body part as a cloud of points where each particle corresponds to a pixel. Occlusions between similarly coloured objects are handled using the exclusion principle [Due00a]. This method penalizes the particles when the objects are close, so that filters do not track the same skin object. The lack of shape information make this algorithm unstable since face and hands filters can be exchanged during the tracking.

In order to address these problems, we propose to use an image template of the subject in addition to shaped model, i.e. a rectangle R_T divided in two regions of equal area (Figure 1(a)). R_{int} and R_{ext} define the sign of the pixel colour probability. Thus the weighted sum of skin probabilities inside R_T,

$$\rho_{R_T} = \sum_{\forall(x,y)\in R} R_T(x,y) \cdot p(c(x,y)|skin) \qquad (3)$$

is minimal when most of the pixels with high probability are inside R_{int} , Figure 1(b). This representation of the face seems to be a fair trade-off between robustness and speed. In addition, considering an image template of the face, updated up to time, make the face tracking more robust to occlusions between similarly coloured objects. Objects are directly labelled using a similarity measure to avoid any exchange between the face filter and any other similarly coloured objects present in the frame, e.g. hands, other people, etc.

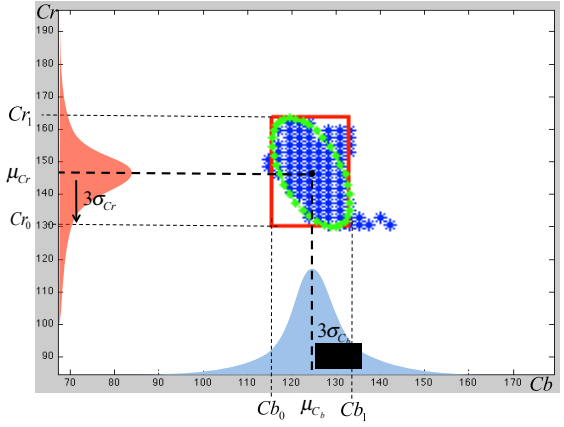


Figure 2: Bivariate normal distribution C_bC_r

A face detection technique using Haar-like features [Vio02a] is used in this work to initialize the model size and the face template. This technique has shown robustness against illumination changes, scale and variation on facial expression for frontal faces. However as soon as the face is fully or partially occluded, detection tends to fail.

The skin model used to compute the skin probability map is built by using the pixels belonging to the face. First we use Kovac *et al* [Kov03a] explicitly defined model in the RGB colour space to extract a rough skin sample region from face. Then we transfom sample pixels into the YC_bC_r colour space and we use them to estimate the mean vector μ_S and the covariance matrix Σ_S Eq.4, of the bivariate normal distribution C_bC_r (Figure 2) that will be used to generate the skin probability map in the next frames.

$$\mu_S = \begin{bmatrix} \mu_{C_b} \\ \mu_{C_r} \end{bmatrix}, \qquad \Sigma_S = \begin{bmatrix} \sigma^2_{C_b} & \sigma_{C_bC_r} \\ \sigma_{C_bC_r} & \sigma^2_{C_r} \end{bmatrix} \qquad (4)$$

4 OBSERVATION MODEL

The entire set of visible features can be used as observation to compute weights. However, it is wiser to select few features that characterize the target among other objects in the frame. We propose to use multiple cues; colour, shape and appearance. Face model gives two measurements for weights computation. Firstly the weighted sum of skin probabilities ρ inside R_T and secondly the *NCC* between the template and the image. The smallest ρ_t^n and the largest NCC_t^n leads to the largest likelihood function between the object model and the observation at the hypothetical state n.

4.1 Colour and Shape Features

The first observation measurement takes into account a rectangular shaped skin blob. First a specific model is

(a) (b) (c)

Figure 3: Tracking results before occlusion (a), during occlusion (b) and after occlusion (c). Big circles represent tracking results of our approach and small circles results of a single feature tracking.

built using the pixels samples from the model initialization step. Colorspace YC_bC_r is used to compute the skin probability map $\mathbf{S}$ by only considering the chrominance components and avoid illumination changes influence. Let $\mathbf{c}(x,y)$ be the colour vector of the pixel at the coordinates (x,y) and $p(\mathbf{c}(x,y)|skin)$ the probability of $\mathbf{c}(x,y)$ to belong to the skin colour class. Thus the first measure ρ_t^n for a particle sample $\mathbf{s}_t^n = x,y$ is expressed as

$$\rho_t^n = \sum_{(x',y')\in R} f_s(x+x',y+y')p(\mathbf{c}(x+x',y+y')|skin) \tag{5}$$

$$f_s(x,y) = \begin{cases} -1 \text{ if } (x,y) \in R_{int} \\ 1 \text{ if } (x,y) \in R_{ext} \end{cases} \tag{6}$$

In order to speed up the algorithm and achieve real time, ρ_t^n is computed using integral images [Vio02a] of the skin probability map. An integral image is an intermediate image representation allowing fast rectangular feature computation. Let $S_{(x,y)}$ be the pixel intensity in the skin probability map S at the coordinates (x,y). The value of the integral image II at (x,y) corresponds to the sum of $S_{(x,y)}$ and all pixels above and to the left. It is expressed as

$$II_{(x,y)} = \sum_{i=0}^{x}\sum_{j=0}^{y} S_{(i,j)} \tag{7}$$

Using this representation any rectangular region can be easily computed performing basic mathematical operations. Let R_i be a rectangle defined by (x_1,y_1) and (x_2,y_2), the sum of the pixels inside the rectangle is computed using Eq.8 and ρ_t^n is easily computed for each particle, Eq.9.

$$R_i = II_{(x_2,y_2)} + II_{(x_1,y_1)} - II_{(x_2,y_1)} - II_{(x_1,y_2)} \tag{8}$$

$$\rho_t^n = R_{ext} - R_{int} \tag{9}$$

This measurement implicitly considers geometric information since the best hypothetical state (particle) correspond to a maximum of skin pixels inside R_{int}.

4.2 Appearance Feature

The second measurement introduces spacial distribution information that is not considered before. This allows to determine where the face is when several skin objects are in the frame. Several measurements can be used to determine similarity between two objects. In this work, we use the Normalized Cross Correlation (NCC) for each hypothetical state (x,y). It is defined as

$$NCC_t^n(x,y) = \frac{\sum_{x',y'} T(x',y')I(x+x',y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 \sum_{x',y'} I(x+x',y+y')^2}} \tag{10}$$

where I represents the image and T the face template. When NCC is closer to 1 the best correspondence between I and T is achieved.

4.3 Multiple Features Observation

Using colour cue to track face and hands has shown good results in a controlled environment. However additional work is required to label each object, e.g. anatomical models. On the other hand tracking by matching a template might be time-consuming depending on the way of optimization. In this paper we propose to combine both features directly in particle filter weight computation. This has as advantage that the skin object is directly labelled by matching the template and no further work is required to distinguish face from hands and vice versa. In addition using the NCC in particle filter is explained, in other words, as an image registration with particle filter optimization which is not time consuming, also integral image representation is used to speed up the algorithm.

Considering colour and appearance measurements, particle weights are defined as

$$\pi_t^n = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(k-\rho_t^n/NCC_t^n)}{2\sigma^2}} \tag{11}$$

where the parameter σ ensures the effectiveness and diversity of particle resampling and k corresponds to a normalization term.

Figure 4: Tracking results of our method (big circle) and tracking results considering a single feature (small circles).

Figure 5: Tracking results of our method (big circle) and tracking results considering a single cue (small circles).

5 EXPERIMENTAL RESULTS

In this section we present the conducted experiments in order to show the robustness of our tracker. A data set containing difficult tracking cases is composed of 4500 frames. It shows several occlusions between the head and the hands, another person and other objects. Firstly we compare our contribution against a model considering only colour cue. Secondly we implement our head tracker in the tracking system proposed in [Gia09a] to show the improvements that our tracker offers to the system.

In the first evaluation framework three experiments are conducted to show difficult tracking cases. The first scenario (Exp. 1) shows a subject holding a city map. The subject fully occludes the face with the map. Figure 3 shows tracking results for the first scenario. Before the first occlusion both methods show similar results. When the head is completely occluded by the map both methods try to reach another skin coloured region. When the head is visible again our method returns to track the head while the other method stays in a local maximum.

Method	Exp. 1	Exp. 2	Exp. 3
Single feature approach	32.9	74.1	34.3
Proposed method	86.89	96.73	76.6

Table 1: GTR(%) for three experiments. Exp. 1: the subject fully occludes the face with a city map. Exp. 2: the subject passes his hands over the head in several directions and various speeds. Exp. 3: other person passes in front of the subject.

The second scenario (Exp. 2) shows the same subject passing his hands over the face. The proposed method gives good results as long as a part of the face is still visible. For example when both hands fully occludes the face, the results is influenced by the low similarity measurement. The lack of shape information make the other algorithm less robust and track the hands instead the face. Figure 4 shows tracking results of series of occlusion.

The third scenario (Exp. 3) shows a person walking through the scene and fully occluding the head of the subject. The proposed tracker follows the subject face whereas the colour based tracks the other person (Figure 5). This case show the robustness of the tracking by adding an image template against similar objects tracking.

Table 1 shows quantitative results of Good Tracking Rate (GTR) for the three experiments described above. The low rate of the colour based method is due to the attachment of the filter to local maximum during or after occlusions. Our filter achieves better results since the local maximum are avoid by considering subject information in addition to colour and shape.

Figure 6: Tracking results of Gianni *et al.* [Gia09a] (top line) and our method (bottom line). Red circle stands for head trackers, blue and green ones for hands.

In the second evaluation framework we have implemented our head tracker in the system proposed by Gianni *et al.* in [Gia09a]. They models each body part (head and both hands) as a cloud of points and allows filters to interact between them to avoid tracking the same skin object. The same principle is considered in our case but we replace the filter that tracks the head with the proposed method. We conducted the evaluation using a corpus where native signers perform deaf sign language, LS-COLIN corpus [Bra01a], to evaluate the robustness of the proposed approach against occlusions and high dynamics within a real context.

The evaluation has been performed using videos of size 320x240 and 720x576. The number of particles for Gianni *et al.* tracker depends on the size of the skin region to track thus to the size of the video. We have chosen for the first video 1750 particles for each hand filter and 3500 particles for the head and the double number of particles for the second video sequence. For the proposed tracker we have chosen 3500 particles for the hands and 1500 particles for the head for both video sizes. Figure 6 shows tracking results of sequences with complex occlusions; two hands occluding the face.

Results from our tracker for this sequence show better performances that the tracker in [Gia09a] since head tracker do not exchange with any hand filter. Table 2 shows the evaluation results for head and hand for two sequences of about 3500 frames. Robust head tracking gives already better results for hands tracking.

Method	Seq.1		Seq.2	
	Head	Hands	Head	Hands
Gianni *et al.*	90.78	88.3	74.2	79.4
Proposed method	99.9	96.65	99.6	96.7

Table 2: Tracking evaluation. Tracking the face robustly arises the quality of the results also for hands tracking.

We conducted our experiments in a laptop Intel Core i7-5500U CPU, 2.4 GHz and 16 GB of RAM. The first evaluation with only the head tracker runs at 30 frames per second, and the second with the hands and head trackers runs at 3.3 frames per second with full resolution images. Hand tracker doesn't have any particulare code optimization, that is why it takes lots more time to process frames beside the high number of particules used per hand (3500×2). The overall evaluation framework validates the proposed method, shows the robustness of the approach and the improvements of implementing it in a tracking system.

6 CONCLUSION AND PERSPECTIVES

In this paper we have addressed real-time head tracking by integrating global information from an image template into particle filtering. We propose an improved particle filter based algorithm for efficient and robust head tracking. Stability is increased when other skin regions are present in the frame. The tracking is performed using the colour and the similarity measure between a template belonging to the model and each particle sample. Since the method uses colour cue and image template of the object to track, the proposed algorithm can be used in many tracking applications. However when head rotation is out-of-plane the similarity measure may by low even at the optimal position.
In future works we intend to make our method template adaptive so that the template can be updated on time. This method will be integrated in our semi-automatic

annotation framework for sign language corpora to enhance sign temporal segmentation and glossing.

7 REFERENCES

[Bir98a] Birchfield,S. Elliptical head tracking using intensity gradients and color histograms. In 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings. pp.232-237, 1998.

[Bla98a] Black,M., and Jepson,A. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In Computer Vision ECCV'98, pp.909-924, 1998.

[Bra01a] Braffort,A., Cuxac,C., Choisier,A., Collet,C., Dalle,P., Fusellier,I., Gherbi,R., Jausions,G., Jirou,G., Lejeune,F., et al. Projet LS-Colin. Quel outil de notation pour quelle analyse de la LS ? In Journées Recherches sur la langue des signes. UTM, Le Mirail, Toulouse, 2001.

[Bra98a] Bradski.G. Real time face and object tracking as a component of a perceptual user interface. In Fourth IEEE Workshop on Applications of Computer Vision, 1998. WACV'1998, pages pp.214-219.1998 .

[Due00a] Duetscher,J., Blake,A., and Reid,I. Articulated body motion capture by annealed particle filtering. In IEEE Computer Society Computer Vision and Pattern Recognition, vol.2, pp. 126-133. 2000.

[Gia09a] Gianni,F., Collet,C., and Dalle,P. Robust tracking for processing of videos of communications gestures. In Gesture-Based Human-Computer Interaction and Simulation, LNAI 5085, Springer-Verlag, pp.93-101, 2009.

[Gon11a] Gonzalez,M., and Collet,C. Robust body parts tracking using particle filter and dynamic template. In 18^{th} IEEE International Conference on Image Processing (ICIP 2011), IEEE, Brussels, Belgium, pp 529–532, 2011

[Gre05a] Greiffenhagen,M., Ramesh,V., and Comaniciu,D. Statistical modeling and performance characterization of a real-time dual camera surveillance system. Apr. 22 2005. US Patent App. 11/112, 930.

[Hag02a] Hager,G., and Belhumeur,P. Efficient region tracking with parametric models of geometry and illumination. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(10): pp.1025-1039, 2002

[Ima98a] Imagawa,K., Lu,S., and Igi,S. Color-based hands tracking system for sign language recognition. In Third IEEE International Conference on Automatic Face and Gesture Recognition, pp.462-467. 1998.

[Isa98a] Isard,M., and Blake,A. Condensation-conditional density propagation for visual tracking. In International journal of computer vision 29, pp.5-28, 1998.

[Jan02a] Jang,D., Jang,S., and Choi,H. 2D human body tracking with structural kalman filter. In Pattern Recognition, 35(10): pp.2041-2049, 2002.

[Kir02a] Kiruluta,A., Eizenman,M., and Pasupathy,S. Predictive head movement tracking using a kalman filter. In IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 27, pp.326-331, 2002.

[Kov03a] Kovac,J., Peer,P., and Solina,F. Human skin color clustering for face detection. In EUROCON International Conference on Computer as a Tool, vol.2, pp.144-148. 2003.

[Mic04a] Micilotta,A., and Bowden,R. View-based location and tracking of body parts for visual interaction. In Proc. of British Machine Vision Conference, volume 2, pp.849-858. 2004.

[Mit07a] Mitra,S., and Acharya,T. Gesture recognition: A survey. In IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37, pp.311-324, 2007.

[Num03a] Nummiaro,K., Koller-Meier,E., and Van Gool,L. An adaptive color-based particle filter. In Image and Vision Computing, 21(1): pp.99-110, 2003

[Pia10a] Piater,J., Hoyoux,T., and Du,W. Video analysis for continuous sign language recognition. In Proceedings of 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, pp.22-23. 2010.

[Rad06a] Raducanu,B., and Vitrià,Y. A robust particle filter-based face tracker using combination of color and geometric information. In Image Analysis and Recognition, pp.922-933, 2006.

[Smo14a] Smolyanskiy,N., Huitema,C., Liang,L., and Anderson,S.E. Real-time 3D face tracking based on active appearance model constrained by depth data. In Image and Vision Computing 11, vol.32, pp.860-869, 2014

[Ste01a] Stenger,B., Mendonça,P., and Cipolla,R. Model-based hand tracking using an unscented kalman filter. In Proc. British Machine Vision Conference, volume 1, pp.63-72. 2001.

[Uri15a] Uřičář,M., Franc,V., Thomas,D., Sugimoto,A., and Hlaváč,V. Real-time multi-view facial landmark detector learned by the structured output SVM. In 11th IEEE International Conference and Workshops on Automatic Face and

Gesture Recognition (FG 2015), Ljubljana, pp. 1-8. 2015.

[Vio02a] Viola,P., and Jones,M.: Robust real-time object detection. In International Journal of Computer Vision 57, pp.137-154, 2002.

[You10a] YoungJoon,C., Seung Ho,S., Kyusik,C., and TaeYong,K Real-time user interface using particle filter with integral histogram. In IEEE Transactions on Consumer Electronics 56, pp.510-515, 2010.

[Zha07a] Zhao,L., and Tao,J. Fast facial feature tracking with multi-cue particle filter. In International Conference on Image and Vision Computing. Hamilton, New Zealand. pp.7-12, 2007.

[Zha13a] Cai,Z.W., Wen,L.Y, Cao,D., Lei,Z., Yi,D., and Li,S.Z.: Person-specific face tracking with online recognition. In 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013, Shanghai, pp.1-6. 2013.

[Zho10a] Zhou,M., Liang,L., Sun,J., and Wang,Y. AAM based face tracking with temporal matching and face segmentation. In 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, pp.701-708. 2010