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

CSRN 2802

(Eds.)

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

Computer Science Research Notes

26. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2018
Plzen, Czech Republic
May 28 – June 1, 2018

Proceedings

WSCG 2018

Short Papers Proceedings

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

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

CSRN 2802

(Eds.)

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

Computer Science Research Notes

26. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2018
Plzen, Czech Republic
May 28 – June 1, 2018

Proceedings

WSCG 2018

Short Papers Proceedings

Vaclav Skala - UNION Agency

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

Computer Science Research Notes
CSRN 2802

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

Published in cooperation with the University of West Bohemia
Univerzitní 8, 306 14 Pilsen, Czech Republic

ISSN 2464-4617 (Print) ***ISSN 2464-4625*** (CD/DVD)
ISBN 978-80-86943-41-1 (CD/-ROM)

WSCG 2018

International Program Committee

Benes,B. (United States)
Benger,W. (United States)
Bouatouch,K. (France)
Bourke,P. (Australia)
Daniel,M. (France)
de Geus,K. (Brazil)
Dingliana,J. (Ireland)
Durikovic,R. (Slovakia)
Feito,F. (Spain)
Feng,J. (China)
Ferguson,S. (United Kingdom)
Galo,M. (Brazil)
Garcia Hernandez,R. (Germany)
Gavrilova,M. (Canada)
Giannini,F. (Italy)
Gudukbay,U. (Turkey)
Juan,M. (Spain)
Klosowski,J. (United States)
Lobachev,O. (Germany)
Molla,R. (Spain)
Montrucchio,B. (Italy)
Muller,H. (Germany)
Patow,G. (Spain)
Pedrini,H. (Brazil)
Renaud,C. (France)
Richardson,J. (United States)
Rojas-Sola,J. (Spain)
Sanna,A. (Italy)
Santos,L. (Portugal)
Segura,R. (Spain)
Skala,V. (Czech Republic)
Sousa,A. (Portugal)
Szecsi,L. (Hungary)
Teschner,M. (Germany)
Thalmann,D. (Switzerland)
Trapp,M. (Germany)
Wuensche,B. (New Zealand)
Wuethrich,C. (Germany)
Xu,K. (China)
Yin,Y. (United States)

WSCG 2018

Board of Reviewers

Aburumman,N. (France)
Assarsson,U. (Sweden)
Ayala,D. (Spain)
Azari,B. (Germany)
Benes,B. (United States)
Benger,W. (United States)
Bouatouch,K. (France)
Bourke,P. (Australia)
Carmo,M. (Portugal)
Carvalho,M. (Brazil)
Daniel,M. (France)
de Geus,K. (Brazil)
De Martino,J. (Brazil)
de Souza Paiva,J. (Brazil)
Dingliana,J. (Ireland)
Durikovic,R. (Slovakia)
Feito,F. (Spain)
Feng,J. (China)
Ferguson,S. (United Kingdom)
Galo,M. (Brazil)
Galo,M. (Brazil)
Garcia Hernandez,R. (Germany)
Garcia-Alonso,A. (Spain)
Gavrilova,M. (Canada)
Gdawiec,K. (Poland)
Giannini,F. (Italy)
Goncalves,A. (Portugal)
Gudukbay,U. (Turkey)
Hernandez,B. (United States)
Horain,P. (France)
Charalambous,P. (Cyprus)
Juan,M. (Spain)
Kanai,T. (Japan)
Klosowski,J. (United States)
Kurt,M. (Turkey)
Lee,J. (United States)
Lisowska,A. (Poland)
Lobachev,O. (Germany)
Luo,S. (Ireland)
Marques,R. (Spain)
MASTMEYER,A. (Germany)
Metodiev,N. (United States)
Molla,R. (Spain)
Montrucchio,B. (Italy)
Muller,H. (Germany)
Oliveira,J. (Portugal)
Oyarzun Laura,C. (Germany)
Papaioannou,G. (Greece)
Patow,G. (Spain)
Pedrini,H. (Brazil)
Peytavie,A. (France)
Puig,A. (Spain)
Ramires Fernandes,A. (Portugal)
Renaud,c. (France)
Ribeiro,R. (Portugal)
Richardson,J. (United States)

Rodrigues,J. (Portugal)
Rojas-Sola,J. (Spain)
Sanna,A. (Italy)
Santos,L. (Portugal)
Segura,R. (Spain)
Skala,V. (Czech Republic)
Sousa,A. (Portugal)
Subsol,G. (France)
Szecsi,L. (Hungary)
Tavares,J. (Portugal)
Teschner,M. (Germany)
Thalmann,D. (Switzerland)
Todt,E. (Brazil)
Tokuta,A. (United States)
Trapp,M. (Germany)
Vanderhaeghe,D. (France)
Vidal,V. (France)
Vierjahn,T. (Germany)
Wuensche,B. (New Zealand)
Wuethrich,C. (Germany)
Xu,K. (China)
Yin,Y. (United States)
Yoshizawa,S. (Japan)
Zwettler,G. (Austria)

WSCG 2018
Short Papers Proceedings
CSRN 2802

Contents

Method of detection similar elements in textures with irregular structure

Vasyl Tereshchenko
Taras Shevchenko National University of Kyiv
64/13, Volodymyrska, st., Kyiv, Ukraine
vtereshch@gmail.com

Yaroslav Tereshchenko
Taras Shevchenko National University of Kyiv
64/13, Volodymyrska, st., Kyiv, Ukraine
y_ter@ukr.net

ABSTRACT

We consider the problem of repetitive elements recognition with heterogeneous image texture. The complexity of solving problems from this class is connected to obtaining precise solution that is sensitive to image with irregular texture. The accuracy depends on correctness of original element selection from which the searching process will start. To solve the problem we propose new synthetic approach that combines statistical methods and machine learning method that allows obtaining resistant and accurate solution. The results of this algorithm are used for detection of reptile skin structure.

Keywords

Pattern recognition, repetitive elements, heterogeneous image structure, computer vision, machine learning

1. INTRODUCTION

Problems of image texture recognition with inhomogeneous and irregular structure are included to the list of problems that haven't a satisfactory solution in general case, becouse it depends on two factors: the accuracy of the selected basic element from which the searching of similar objects starts, and how algorithm adequately responds to the regularity and high level of noise on input image. Known approaches solve the problem only in the case of regular and uniform structure of texture image. However, for some subclasses of these problems, in which elements of image texture divided into similarity classes, can suggest approaches that give satisfactory results. An example of this class is the task of recognizing image elements of the skin structure of reptiles and fish that are generally have inhomogeneous and irregular structure. There is a problem of identification these animals by photo of their skin and thereby identify the class or type of animal. Existing approaches that solve such problems [Leu96a, Fir11a] can be used for identification only regular and uniform structure of texture for further automatic extrapolation search of these elements from a given basic sample. Solving the problem in the case of irregular and non-uniform grid leads to serious mistakes and not reliable results. In particular, if the algorithms use a method Canny [Zho11a] for edge detection, then because of gradients absence in the boundaries of input image the algorithm can not adequately segment "scales" (Figure 1).

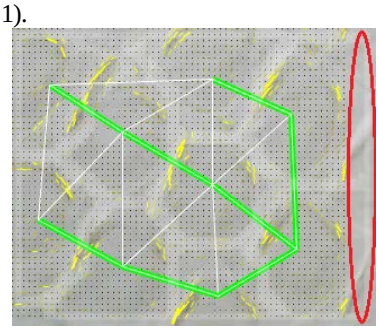


Figure 1. Lack of gradients on the border of the image

Also, existing algorithms [Leu96a –Dud73a] are extremely sensitive to boundaries (gradient changes) in the image, especially if there is additional a lot of noise on it. The extra boundaries may also to appear in case of the defects of animal skin (scars, defects in processing, etc.) and in case of non-uniform background, if the skin does not occupy the whole image (see Figure 2; in this case, a large number of outliers that spoils further work of algorithm).

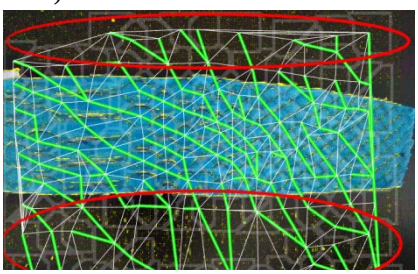


Figure 2. Excessive amounts of noise in the image.

Conventional filtering (smoothing image) does not solve the problem, because there is no testing: is there a noise or a part of the searching element. Excessive smoothing can lead to loss important information. Also, each image has its own noise level, and therefore we can not use the same filtering options. In addition, existing algorithms work incorrectly in the case of large differences in the size and form of these elements [Leu96a]. They inadequately ignore scales (Figure 3), or integrate them into large clusters (Figure 4).

Figure 3. Bonding of elements.

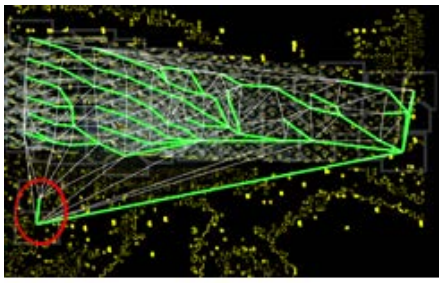


Figure 4. Combining elements of small sizes into large clusters.

There are several approaches to solving this problem. They are usually divided into two types: structural and reference. First approach [Leu96a] is based on separation and analysis of various structural elements and their features, properties that identify the object. Second approach involves comparison of the investigated sample with defined set of templates [Seo10a, Bis11a]. In this paper we propose a new approach to solve the problem of recognition of the similar repeatable elements on image with irregular and heterogeneous structure of texture using the combination of statistical methods and machine learning. This algorithm we used to segment the skin of reptiles and fish scales.

Objective: To develop an efficient algorithm to solve the problem of recognition of similar repeatable elements for non-uniform and irregular structures of image texture.

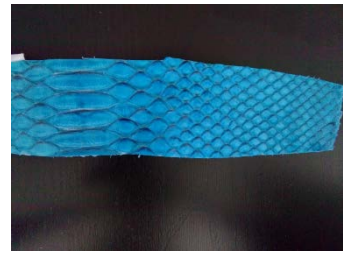
2. METHOD OF DETECTION SIMILAR ELEMENTS

In order to determine similar repeatable elements on inhomogeneous and irregular image texture, we should develop algorithms to solve two main problems: generation initial points and constructing "approximating grid". The quality of the resulting recognition depends on the efficiency and accuracy of these algorithms. Let's consider methods for solving these problems. Before proceeding to develop these algorithms we will perform pre-processing.

2.1 Preprocessing

Simple image has a lot of unnecessary information that can reduce accuracy of the algorithm. For this, we should implement some preprocessing procedures. It is known that every image has target object (for example, skin of reptile) and background. That is why, to separate them, we can make the following steps (Figure 5):

1. Gaussian blur [Dor14a] with kernel value $(7 \times 7, 11 \times 11)$ for 640×480 image.
2. Using k-means clustering [Coa12a] with $k = 2$ execute image binarization.
3. With connected components algorithms, we can detect border of target object [Cha04].



a)

b)

Figure 5. a) Input image; b) separated background

Now we proceed to consider methods of solution mentioned above problems. Let's start from a problem of generation initial points.

2.2 Searching of initial points

In our case, initial point is a centroid of the repeatable element considering its shape. At the

ISBN 978-80-86943-41-1

moment, the most satisfactory shape type is the bounding box. It uses less memory and more flexible for different image processing algorithms. Detection of initial element among various noises (for reptile skin - scratches, pigmentation, high illumination and etc.) isn't a simple problem. That is why, structural methods have some constrains in processing such difficult textures. In our opinion, the most appropriate method - constructing a so-called "probabilistic grid" that can detect elements of texture for generating initial points. The grid involves initialization stage based on machine learning approach (SVM [The09a, Ing08, Vap98a] is used for searching candidates for initial point).

2.2.1 The algorithm of candidates detection for initial element using SVM

1. Generation of training set (each image has its own layout on which negatives and positives are generated). Positive << Negatives
2. Training:
   a) Provided HOG descriptor [Low04, Dal05a];
   b) Normalized on 64 × 64 kernel;
   c) Training with RBF kernel.
3. Recognition:
   a) Pass of the image according to different sizes of sliding windows;
   b) Removing outliers using NMS [Neu06];
   c) Determination of maximum similarity coefficients.

2.2.2 Constructing of "probabilistic grid"

The probabilistic grid works with candidates for initial point, which are generated by SVM. The basic stages of grid generation process are following:

1. Triangulation (Delaunay) of candidates set for the initial point (Figure 6). We can consider p initial points that belong to monotone triangulation (set of monotone chain of triangles).

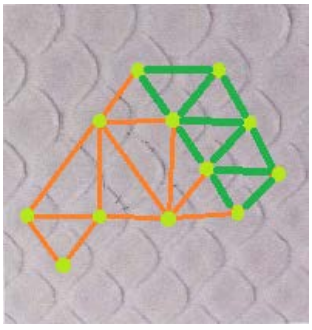


Figure 6. Orange lines–triangulation Delaunay. Green lines – monotone.

2. The selection criterion of initial point: it is the point that has the smallest dispersion of the lengths (Figure 7) of its neighbors and the minimum asymmetry coefficient.

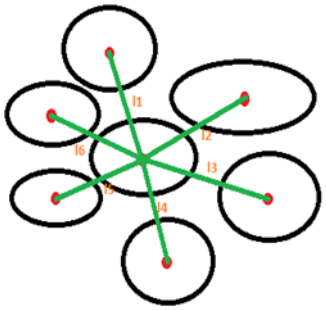


Figure 7. Point neighborhood.

For this, we will use following formulas:

$M_k = \frac{1}{n_k}\sum_{i=1}^{n_k} l_i$ - the mean length of neighbors for point k,

$D_k = \frac{1}{n_k}\sum_{i=1}^{n_k}(l_i - M_k)^2$- dispersion for point k.

Where n_k - the number of neighbors for the current point, l_i- length of i – neighbor.

The asymmetry coefficient is defined as:

$$E_k = \sum_{i=1}^{n_k} \frac{(v_i, v)}{\|v\|} \to min \quad (1)$$

3. After preliminary procedures all points will have own value of dispersion and entropy (2):

$$F_k = (D_k, E_k) \quad (2)$$

Then the initial point is defined as:

$$k_{initial} = \underset{k}{\operatorname{argmax}} \|F_k\|. \quad (3)$$

We can generate p best initial points, using points from monotone triangulation.

2.3 Generating of similarity classes

After determining the initial elements we construct classes of similar elements. For this "approximating grid" are generated. In addition, we use LSK-algorithm [Seo10a] to determine the correlation between images.

2.3.1 Feature Extraction

For feature extraction we use LSK/PCA algorithm [Seo10a]. It consists the following stages:

1. *Finding LSK kernel.* The main component of feature extraction algorithm is analysis of details of image based on kernel function. These details are gradients. In our case, the kernel function is:

$$K = \frac{\sqrt{\det(C_l)}}{2\pi h^2} e^{-\frac{(x_l - x)^T C_l (x_l - x)}{2h^2}}, \quad (4)$$

where C_l- covariance matrix for the gradient in 4 directions. Therefore, each kernel will have a matrix of values of smaller dimension function.

ISBN 978-80-86943-41-1

2. *Normalization LSK features and dimension reducing*. We make normalization for reducing the dispersion of kernel function values. Target image is divided into patches (Figure 8).

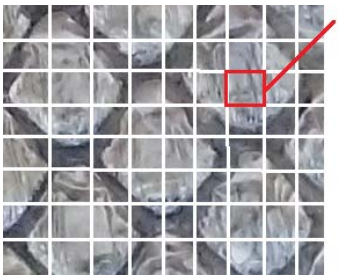


Figure 8. Target decomposition on patches.

Each patch is corresponding to LSK core and normalized by the formula [Seo10a]:

$$W_Q^j = \frac{K_Q^j}{\sum_{l=1}^{P^2} K_Q^j}, l = 1, \dots, P^2, j = 1, \dots, n, \quad (5)$$

where n - number of patches. Reducing the dimension of the features matrix is conducted using PCA algorithm.

3. *Comparison of feature vectors*. We use Frobenius [Seo10a] dot product for comparison of feature vectors, and build resemblance map:

$$\rho_i \equiv \rho(F_Q; F_T) = \sum_{l=1}^{n} \frac{f_Q^{l\,T} f_{T_i}^l}{\|F_Q\|_F \|F_{T_i}\|_F} \quad (6)$$

Resemblance map:

$$f(\rho_i) = \frac{{\rho_i}^2}{1-\rho_i} \quad (7)$$

2.3.2 *Constructing of "Approximating grid"*

This kind of structure can help to create adaptive algorithm for repeatable objects detection. The main procedure of approach is following:

1. Start from initial point 1 (Figures 9).

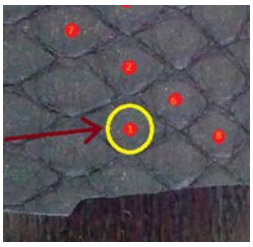


Figure: 9. Location of initial point.

As mentioned in previous paragraph, we can start from multiple initial points. That is why, it is possible to repeat this procedure for every element. It can help to accelerate creation process.

2. For given size of the original element we build probabilistic neighborhood. In this neighborhood, we will look for possible similar elements (Figure 10). The shape of this area can be different. In our case, it is rectangle with sizes that correspond to initial element with empirically estimated scale coefficient.

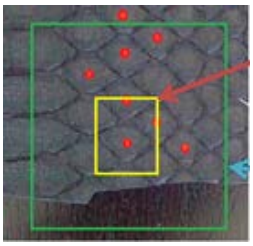


Figure 10. Yellow rectangle – initial template, green rectangle – neighborhood.

3. Changing the size of the pattern, we build correlation map using LSK kernel. For each template we determine the greatest value of correlation function relative to its various proportions (Figure 11.left). If correlated element is located on border of searching area, we extend resemblance map.
4. Find the new center (Figure 11.right).

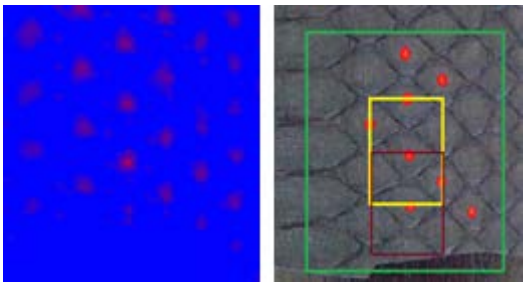


Figure 11. Left- resemblance map; right- new center.

Also, we can generate multiple centers. To merge them, we make clustering by peaks location in resemblance image map. For example, we can use algorithm FOREL [Zag86a] for any number of clusters or k-means for k nearest neighbors. In our case, FOREL is more flexible. Centers of these clusters are new initial points (Figure 12, Figure 13).

5. For created points we repeat steps 1 - 4 and consider new search neighborhood.

After all procedures, we can make the final Delaunay Triangulation for the new set of points and generate grid for whole image texture.

ISBN 978-80-86943-41-1

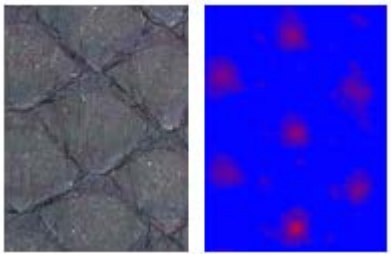


Figure 12. left - input area; right - resemblance map.

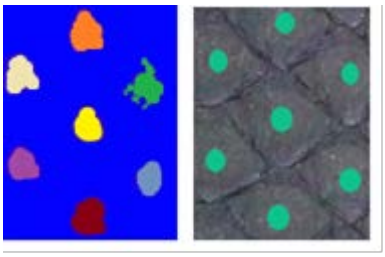


Figure 13. left - FOREL clusters; right- generated points.

3. IMPLEMENTATION

The algorithm is implemented in C ++ using libraries: opencv, libcvm. It was used to the image texture of snake and crocodile skin, Figure 14, Figure 15.

Figure 14. Sample of image fragment.

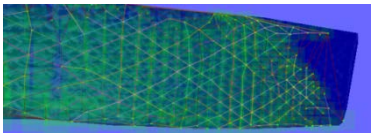


Figure 15. Approximation grid for similar object detection.

Quality of the approach was tested on own database. Using our dataset, we generate precision-recall curve for different initialization methods. In Figure 16. we see that detection algorithm with SVM and "probabilistic grid" has better PR rate than algorithm with only SVM initialization.

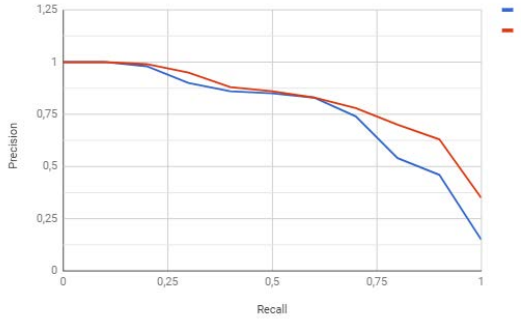


Figure 16. Precision-recall curve. Blue – only SVM, Red – SVM and "probabilistic grid".

The speed of the algorithm depends on number of repeatable elements. For example, image from Figure 14 takes 2.51 sec of processing time on mobile device Samsung Galaxy S6. Figure 17 shows dependence between time and number of elements.

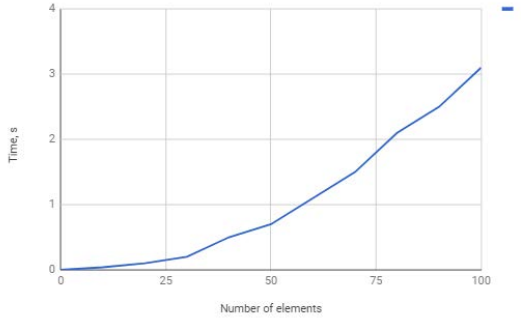


Figure 17. Time dependence.

4. CONCLUSION

We proposed a new approach to solve the problem of finding repeatable objects on the image in case of non-uniform and irregular structure of texture. According the approach we split the problem into two subtasks: search initial element and generation of similar objects. To solve the first subtask we proposed a statistical method that uses a "probabilistic grid" and support vector machine (SVM). To solve the second problem we proposed a new data structure - "approximating grid". Constructing of the grid includes searching of correlation map, which must specify certain characteristics of the target image. Features were generated using LSK descriptor with PCA algorithm that reduces the dimension of vector space. Finding the grid is an iterative process in which each new point generates searching neighborhood. The results of the algorithms we used to develop software for reptile skin segmentation based on texture structure (set of scales). The algorithm accurately finds centers of

ISBN 978-80-86943-41-1

scales and separates them, even when the boundary between the elements is invisible for the human eye.

REFERENCES

[Leu96] Leung, T. , Malik, J. (April 1996). Detecting, localizing and grouping repeated scene elements from image. Fourth Euro. Conf. on Computer Vision, Cambridge, pp. 22-24, 1996.

[Fir11a] Firouzi, H., Najjaran, H. Detection and Tracking of Multiple Similar Objects Based on Color-Pattern. Lecture Notes in Computer Science: Autonomous and Intelligent Systems, pp. Vol. 6752, pp. 273-283, 2011.

[Zho11a] Zhou, P., Ye, W., Wang, Q. An Improved Canny Algorithm for Edge Detection. Journal of Computational Information Systems, 7(5), pp. 1516-1523, 2011.

[Vap74] Vapnik, V., Chervonenkis, A.Y., Theory of Pattern Recognition (in Russian). Nauka, Moskow , 1974.

[Seo10a] Seo, H J., Milanfar P. Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32(9), pp. 1688-1704, 2010.

[Bis11a] Bishop, C. M. Pattern Recognition and Machine Learning. Series: Information Science and Statistics, 2011.

[Dud73a] Duda, R.O., Hart, P. Pattern Classification and Scene Analysis. New York: Wiley International, 2011.

[Dor14a] Dorga, A. and Bhalla, P. Image Sharpening By Gaussian And Butterworth High Pass Filter. Biomedical & Pharmacology Journal, Vol.7(2), pp. 707-713, 2014 .

[Coa12a] Coates, A., Ng, A. Y. Learning Feature Representations with K-means . Neural Networks: Tricks of the Trade, Springer LNCS, Vol. 7700, pp. 561-580, 2012.

[Cha04a] Chang F., Chen, C.J., Lu, C.J. A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique. Computer Vision and Image Understanding, Vol. 93, No.2, pp. 206-220, 2004.

[The09a] Theodoridis, S., Koutroumbas, K.. Pattern Recognition . Academic Press, 2009.

[Ing08] Ingo, S., Andreas, C. Support Vector Machines. New York: Springer-Verlag, 2008.

[Vap98] Vapnik, V.N., Statistical Learning Theory. John Wiley, NY, 1998.

[Low04a] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *IJCV*, Vol. 60(2), pp. 91–110, 2004.

[Dal05a] Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. Procedings *of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA, pp. 886-893, 2005.

[Neu06] Neubeck, A., Van Gool, L. Efficient Non-Maximum Suppression. Proceedings of 18th International Conference on Pattern Recognition, Hong Kong, pp. 850-855, 2006.

[Zag86a] Zagoruiko, N.G., Elkina, V.N., Lbov, G.S., Emelianov, S.V. OTEKS applied software. Moscow: «Finances and statistics», 1986.

 ISBN 978-80-86943-41-1

A fast approximation of the Voronoi diagram for a set of pairwise disjoint arcs

Dmytro Kotsur
Taras Shevchenko National University of Kyiv
64/13, Volodymyrska, st., Kyiv, Ukraine
dkotsur@gmail.com

Yaroslav Tereshchenko
Taras Shevchenko National University of Kyiv
64/13, Volodymyrska, st., Kyiv, Ukraine
y_ter@ukr.net

Vasyl Tereshchenko
Taras Shevchenko National University of Kyiv
64/13, Volodymyrska, st., Kyiv, Ukraine
vtereshch@gmail.com

ABSTRACT

We propose a method for fast approximation of the Voronoi diagram for a set of pairwise disjoint arcs on a plane. The arcs are represented by parameterized curves. A set of input curves is discretized into partition set, for which the Voronoi diagram is constructed. After merging corresponding Voronoi cells and removing redundant edges, the Voronoi graph is approximated by Bezier curves. We also propose the elaboration and optimization of the approximation. The total complexity of the algorithm is $O(N \log N)$ in the worst-case.

Keywords

Approximation, Voronoi diagram, Voronoi cell, Bezier curve, discretization, partition set, parametric curve.

1. INTRODUCTION

Relevance. Today there is a wide variety of algorithms for solving problems of computational geometry. But typically, the scope of these algorithms is very narrow. For example, well-known algorithms for constructing the Voronoi diagram (e.g., "divide and conquer" [Prep85b], Fortunes algorithm [Fort87a]) can be effectively applied only to a set of points and line segments. However, when we solve practical problems, we usually deal with more complex geometrical shapes than just points or line segments. Normally these complex objects can be represented by parametric curves of arbitrary shape [Aic10a].

However, the construction of the exact Voronoi diagram for the set of parametric curves is not a trivial task. Even in the simplest example of the Voronoi diagram for two arbitrary disjoint curves (which is just a bisecting curve between them) we have to consider a large number of particular cases. The construction of the Voronoi diagram for three or more objects represented by parametric curves is even more sophisticated and would require a huge amount of computational time. Therefore, for practical applications it is reasonable to reduce the problem of exact Voronoi diagram construction to a problem of its approximation construction, which can be performed in a reasonable computational time. However, the problem of constructing approximations of Voronoi diagram is not trivial and still requires further study.

Analysis of recent research and publications. In a large amount of works devoted to approximation of Voronoi diagram, authors consider the spatial discretization of 2D plane into cells using a discrete grid with a fixed step [Sud06c], [Hof99c]. In this case, the plane is sampled and then the result (discrete image) is used for further transformations. After performing all necessary transformations, the Voronoi diagram is obtained. A significant drawback of these methods is that the accuracy of the constructed Voronoi diagram depends on the size of the grid, and reducing its size leads to a significant increase in the number of cells (quadratic dependence). Thus, in order to achieve acceptable results in terms of precision, we would require a lot of computational power. Therefore, these methods can be extremely time-consuming in some cases.

Another approach is an approximation by means of constructing the Voronoi diagram for the simplest geometric objects, such as points or line segments. In particular, in [Ho09c] authors demonstrate an approach to approximate the Voronoi diagram for arbitrary geometric objects using Bezier curves and taking into account the Voronoi diagram for points.

However, this approximation is made only for the part the Voronoi diagram (they solve the problem of finding the minimum path). The approximation by means of the Voronoi diagram for points has significant prospects as well as it makes possible to operate with less amount of simple objects. The points on the curves can be selected based on the discretization step, which can be fixed or can depend on the characteristics of the curve. It allows to speed up the construction of approximation, and at the same time, to maintain the desired accuracy in critical regions. Since the method is based on the construction of the Voronoi diagram for simple geometric objects (points or line segments), the critical regions of such approximation can be easily refined by supporting dynamic Voronoi diagram for points and inserting new points (or line segments), where it is necessary.

Also there are some attempts to compute exact Voronoi diagram [Ary02c], [Ary02b], [Har01c]. In paper [Seo08c] authors describe an algorithm for computing the precise Voronoi Diagram of planar freeform curves, which represented with a parametric form. The authors try to build the precise bisector between two curves and precise junction points. But this effort leads to the necessity of solving the system of three nonlinear equations, which is not trivial task itself, since it requires time-consuming numerical methods. Another similar approach is described in [Ram99a]. It has an asymptotic time of $O(n^2)$, where n is a number of curves. This method has the same drawback as previous: computation of junction points (authors use Newton-Raphson method).

Novelty and ideas. The purpose of this paper is to develop an algorithm for fast and accurate approximation of Voronoi diagram, which has $O(N \log N)$ complexity in the worst case. Our approach is based on a point sampling for input curves and further construction of Voronoi diagram for sampled points. Then we use a developed procedure for merging the Voronoi cells and approximate edges of Voronoi diagram by curves.

2. A METHOD FOR FAST APPROXIMATE VORONOI DIAGRAM CONSTRUCTION

Before describing the method and the solution of a problem we recall the basic concepts [Aur13b] used in this paper.

2.1 Basic concepts and statement problem

Definition 1. Suppose we are given a set of generator points $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^{\mathrm{d}}$, where $(2 \le n < \infty)$. We call set P the generator set of the Voronoi diagram. Let's denote by I_n a set of generators indices and Euclidean distance between two objects x and y as $\rho(x, y)$. We call the region given by:

$$VP(p_i) = \{x | \rho(x, p_i) \le \rho(x, p_j)\}, \quad (1)$$

where $(j \ne i), i, j \in I_n, x \in \mathbb{R}^2$, the Voronoi cell (Voronoi polygon) associated with p_i. Then the Voronoi Diagram generated by P (or the Voronoi diagram of P) is defined as follows:

$$VD(P) = \{VP(p_1), VP(p_2), \dots, VP(p_n)\} \quad (2)$$

For any two generators p_i and p_j we define a region of dominance of p_i over p_j:

$$H(p_i, p_j) = \{x | \rho(x, p_i) \le \rho(x, p_j), x \in \mathbb{R}^d\}, \quad (3)$$

where $i, j \in I_n$. Thus, the Voronoi cell can be defined by the following statement:

$$VP(p_i) = \bigcap_{j \in I_n \setminus \{i\}} H(p_i, p_j) \quad (4)$$

The boundary or bisector between two regions of dominance $H(p_i, p_j)$ and $H(p_j, p_i)$ is denoted by $b(p_i, p_j)$ and defined as follows:

$$b(p_i, p_j) = H(p_i, p_j) \cap H(p_j, p_i), \quad (5)$$

or alternatively:

$$b(p_i, p_j) = \{x | \rho(x, p_i) = \rho(x, p_j), x \in \mathbb{R}^d\} \quad (6)$$

For a given generator set P and a set of indexes I_n, the boundary $b(p_i, p_j)$ can be denoted in a short form as $b_{i,j}$.

Definition 2. Let $C(t) = (x_c(t), y_c(t))$ be a continuous parametric curve on a plane, $t \in [0,1]$, and parameter Δt determines the step of approximation. We call set of points $P_c = \{\tilde{c}(i\Delta t) | \, i = \overline{0, n}\}$ the partition set of the curve C, where $n = \left[\frac{1}{\Delta t}\right]$.

Problem statement.

Let $\mathcal{C} = \{C_1(t), C_2(t), \dots, C_n(t)\}$ be a set of continuous parametric curves, which are pairwise disjoint. Given the partition sets $P_{c_1}, P_{c_2}, \dots, P_{c_n}$ of curves $\mathcal{C}$ and their union $\mathcal{P} = P_{c_1} \cup P_{c_2} \cup \dots \cup P_{c_n}$, build an approximation of Voronoi diagram $VD(\mathcal{C})$ for set of curves $\mathcal{C}$.

2.2 The solution for arbitrary objects

At the first we build Voronoi diagram for union of partition points $\mathcal{P}$:

$$VD(P) = \{VP(p) | p \in P\} \quad (7)$$

Let $l: P \to \mathbb{N}_+$ be the function, that for a given point returns index of curve, which this point belongs to: $l(p) = i \overset{def}{\Longleftrightarrow} p \in P_{c_i}$. Then the approximation of Voronoi cell associated with curve C_i is obtained by the union the Voronoi cells for each point in the corresponding partition set:

$$\widetilde{VP}(C_i) \cong \bigcup_{l(p)=i} V(p) \quad (8)$$

Thus, the resulting approximation of the Voronoi diagram for the set $\mathcal{C}$ is:

$$\widetilde{VD}(\mathcal{C}) \cong \{\widetilde{VP}(C_i) | i = \overline{1, n}\} \quad (10)$$

ISBN 978-80-86943-41-1

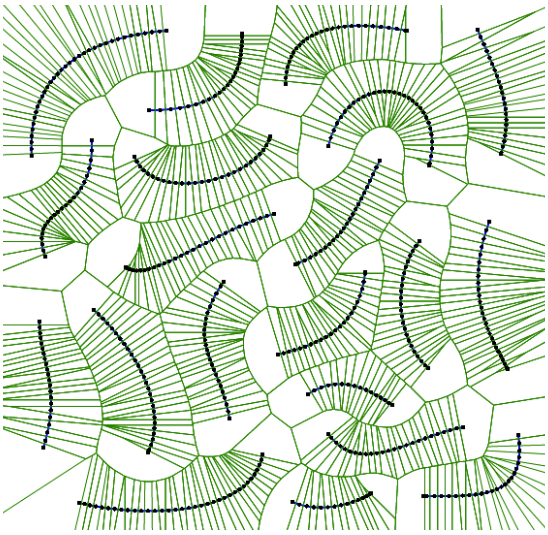


Figure 1. Voronoi diagram (green) for a set of sampled points (black) comprising the partition set of input nonintersecting Bezier curves (blue)

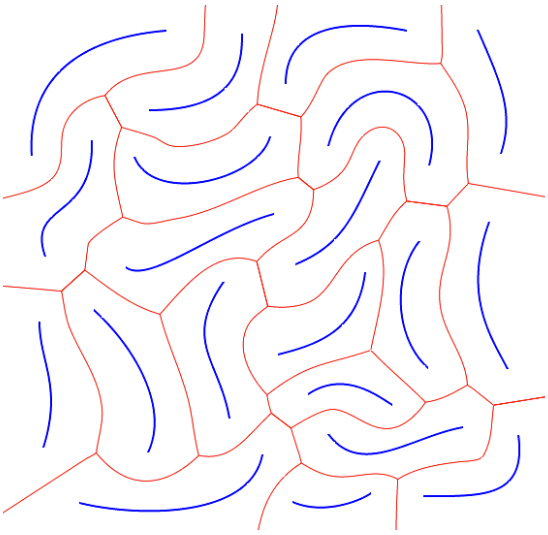


Figure 2. An example of approximate Voronoi diagram (red) for a set of 20 nonintersecting Bezier curves (blue)

Initially, we choose a certain set of curve points for constructing a Voronoi diagram, while maintaining the connection of each point with the corresponding curve. Next, we merge together the Voronoi cells, whose centers belong to the same curve, and remove the adjacent edges.
Figures 1 and 2 show an example of constructing a Voronoi diagram for 20 nonintersecting Bezier curves.

2.3 Approximating Voronoi edges with curves

After constructing the Voronoi diagram for points and merging corresponding cells (removing respective edges), we obtain an approximation by polygonal chains (each chain connects two junction points). At the next step we approximate obtained chains with curves.

At first to make the approximation by curves we should choose the canonical equation of a curve. The type of approximating curve (canonical equation) for Voronoi edge depends on the type and parameters of curves, which it separates. We consider the general case and make approximation by quadratic and cubic Bezier curves. Other types of curves may be similarly considered.

One of the most appropriate methods of approximation by curves is least square approximation. We fix the first and last points (start and end point) of Bezier curve and then use the least squares method to find best curve fit.

Thus, for quadratic Bezier curves we find the coordinates of point P_1:

$$B(t) = t^2P_0 + 2(1-t)tP_1 + (1-t)^2P_2 \qquad (11)$$

In this case the least square method is reduced to the solution of a linear equation with one variable for each coordinate, which can be easily solved:

$$\varphi(P_1) = \sum_{i=1}^{n}\left(P_i^* - B(t_i)\right)^2 \to \min \Longrightarrow \frac{d\varphi}{dP_1} = 0 \quad (12)$$

In order to approximate polyline with a cubic Bezier curve we should find x, y coordinates of two points P_1 and P_2:

$$B(t) = t^3P_0 + 3(1-t)t^2P_1 + 3(1-t)^2tP_2 + t^3P_3 \quad (13)$$

This problem reduces to the solution of the system of linear algebraic equations for each coordinate. Each system of equations consists of two equations:

$$\varphi(P_1, P_2) \to \min \Longrightarrow \begin{cases} \dfrac{d\varphi}{dP_1} = 0, \\ \dfrac{d\varphi}{dP_2} = 0. \end{cases} \qquad (14)$$

where

$$\varphi(P_1, P_2) = \sum_{i=1}^{n}\left(P_i^* - B(t_i)\right)^2 \qquad (15)$$

Thus, the total number of equations in the system of linear equations depends on the order Bezier curve.

Implementation details. For each type of approximation curve we get an analytical solution (expressions for each of unknown point), which is easy to implement in code.

3. COMPLEXITY ANALYSIS

The analysis of the complexity of the proposed method is provided in the following statements.

Theorem 1. If an input set consists of m objects on a plane represented by nonintersecting parametric curves and the total number of points used to discretize these m objects is N. Then, approximation of Voronoi

ISBN 978-80-86943-41-1

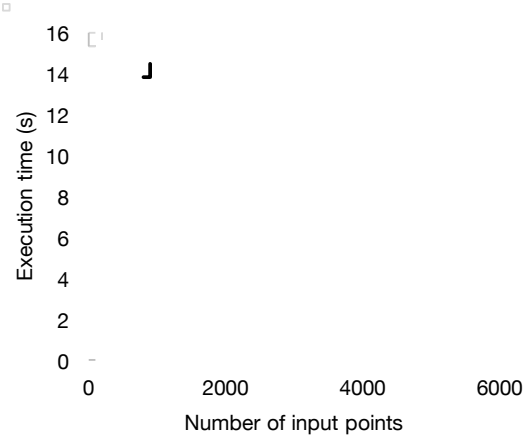


Figure 3. Execution time of the proposed method for Voronoi diagram approximation (x-axis is a number of input objects, y-axis shows an execution time in seconds)

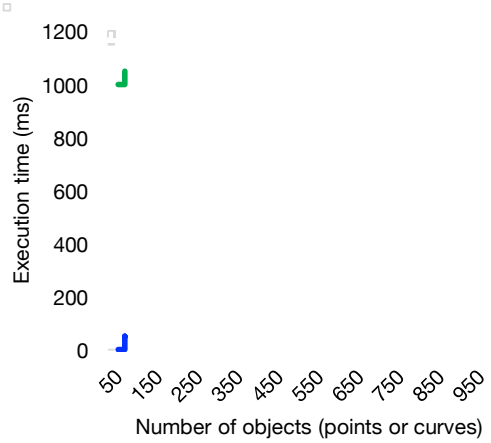


Figure 4. Execution time comparison: green curve is proposed method for an approximate Voronoi diagram construction for *N* curves; blue curve shows execution time of "divide and conquer" [Prep85b] algorithm for *N* points on a plane.

diagram for the set of m arbitrary-shaped objects represented by parametric curves, can be computed in time $O(N \log N)$.

Proof. In papers [Prep85b, Fort87a, Sha75c] authors provide the detailed description and complexity analysis of an algorithm for a Voronoi diagram construction for a set of N points in $O(N \log N)$. If Voronoi diagram is represented by doubly connected linked list, then the following lemmas hold:

Lemma 1. Merging two neighboring Voronoi cells represented by doubly-connected linked lists can be performed in $O(1)$ time.

Proof. In order to merge two neighboring Voronoi cells we should merge corresponding doubly-connected linked lists. This operation is simple pointers reassignment and it can be performed in $O(1)$ time.

Lemma 2. Approximation of the Voronoi diagram for arbitrary-shaped parametric curves on a plane can be performed using the pre-computed Voronoi diagram for points in $O(N)$ time.

Proof. An approximation of Voronoi diagram is performed by merging the neighboring Voronoi cells, whose generators correspond to the same curve. The maximal number of Voronoi cells is N and one pair of neighboring cells can be merged $O(1)$. Thus, we can get an approximation of Voronoi diagram with edges represented by polylines in time $O(N)$ (by merging all necessary pairs of cells).

Taking into account Lemmas 1, 2 and the following statement: quadratic or cubic Bezier curves fit polynomial chains in time $O(M)$, where M - number of points in chain; we can formulate following lemma:

Lemma 3. An approximation of Voronoi diagram for arbitrary-shaped objects on a plane using Bezier curves can be computed in $O(N \log N)$ time.

Therefore, the approximation of the Voronoi diagram for a set of m arbitrary-shaped objects on a plane, which are represented by non-intersecting parametric curves can be performed in time $O(N \log N)$, that concludes the proof.

4. IMPLEMENTATION DETAILS

In the implementation part we constructed the partition for a set of curves (based on the uniform point sampling) and then build the Voronoi diagram for the obtained partition using the "divide and conquer" algorithm described in [Sha75c]. In order to store the correspondence between curve indexes and points in partitioning, we used a hash map. For every index of a point it stores the index of the corresponding curve and also index of previous and next sampled point on a

Point Index	Next point index	Previous point index	Curve Index
1	2	-1	1
2	3	1	1
3	4	2	1
…	…	…	…
N_1	-1	N_1-1	1
N_1+1	N_1+2	-1	2
N_1+2	N_1+3	N_1+1	2
…	…	…	…
N	-1	N-1	M

Table 1. Example of a hash table, which maps point indices to curve indices, it also stores indices of previous and next points on a curve;

ISBN 978-80-86943-41-1

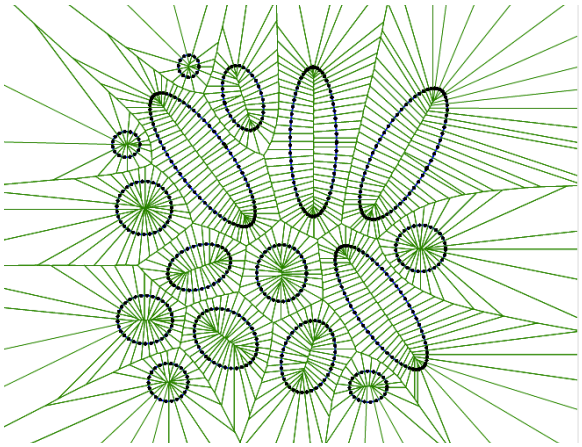


Figure 5. Voronoi diagram for a set of sampled points comprising the partition set of input curves

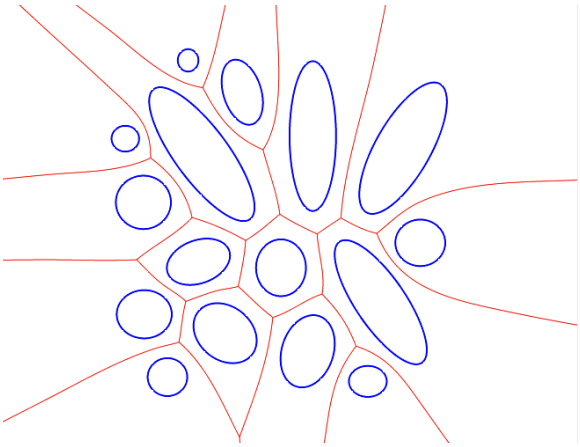


Figure 6. An approximate Voronoi diagram (red) for a set of 16 ellipses (blue)

curve (for example, see Table 1, value -1 indicates no data). Another hash table maps index of a curve to an index of one of its endpoints from partition set. At the next step we merge Voronoi cells for neighboring points of a curve and get an approximate Voronoi cell of a curve (whose edges are polygonal chains). Voronoi diagram is represented by doubly-connected edge list (DCEL). The procedure of merging is the following: we start from some partition point of curve p; run BFS(p) and iterate through all neighboring points of the same curve. At each step we merge pair of Voronoi cells corresponding to the neighboring points and remove redundant edges of Voronoi cells. During the procedure of merging we also determine junction points of the resulting Voronoi diagram.

5. EXPERIMENTAL RESULTS

The practical implementation is made in C++ using OpenGL graphic library to visualize data and Qt framework for GUI. An input of data is provided either by user manually or from SVG-files. The implemented code allows also to visualize the main stages of our algorithm (see Figures 1-2, 5-9). We also tested the performance of our method. All experiments in this paper were carried on Intel Core i7 2.3GHz processor computer with 4GB RAM.

Figure 3 illustrates the results of the execution time testing. The execution time of the proposed algorithm was compared to the execution time of "divide and conquer" algorithm as described in [Prep85b]. This comparison (see Figure 4) shows how the complexity of input objects influences the computational efficiency of the method.

Figure 4 also demonstrates the increase in computational time for curves in comparison to a set of points by the factor of approximately 20 (in case of the discretization step equal to ~0.1). The main reasons for such increase are computational overheads (curve discretization, approximation) and increase in total number of processed points.

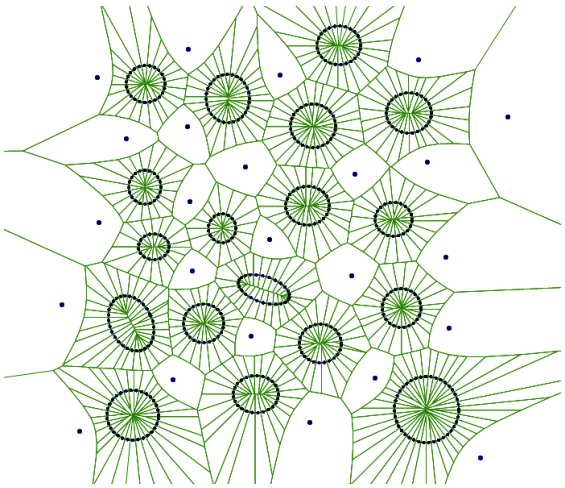


Figure 7. Voronoi diagram for a set of sampled points, which comprises a partition set

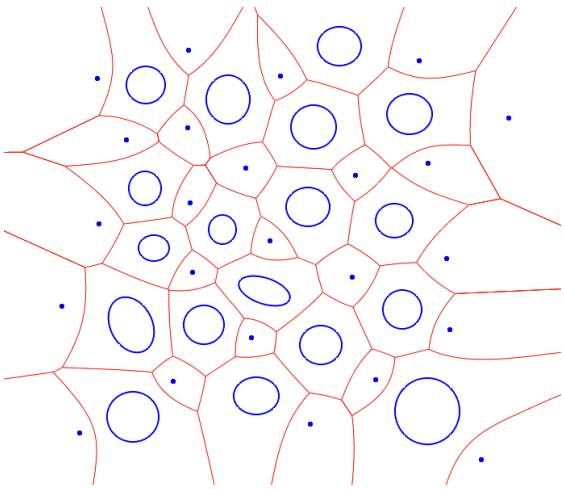


Figure 8. An approximate Voronoi diagram (red) for a set of 18 ellipses and 24 points (blue)

ISBN 978-80-86943-41-1

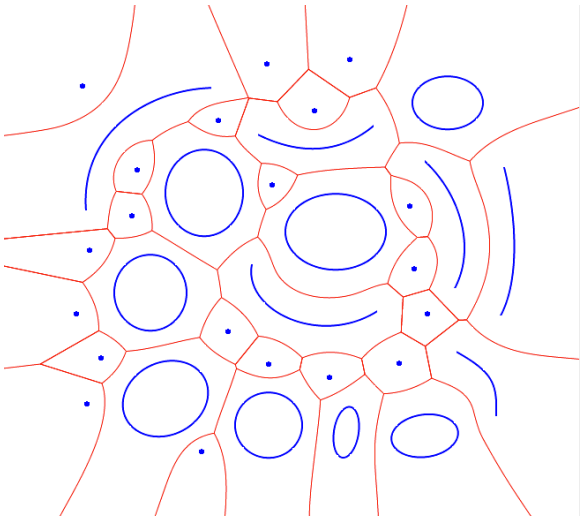


Figure 9. An example of approximate Voronoi diagram (red) for a set of 6 nonintersecting Bezier curves, 8 ellipses and 20 points (blue)

6. CONCLUSION

Thus, we propose an algorithm for approximation of the Voronoi diagram for a set of pairwise disjoint arcs on a plane. Arcs are represented by parametric curves. For curves we construct a partition set for which the Voronoi diagram is built. At the next step we perform a transformation of the obtained Voronoi diagram by merging neighboring Voronoi cells, which correspond to the same curve, and removing unnecessary edges. Thus, we obtain an approximation of Voronoi diagram with polynomial chains. We also propose to approximate these polygonal chains by Bezier curves and arcs. Cases of cubic and quadratic Bezier curves were analyzed. The type of approximating curve is chosen analytically. The total complexity of the proposed algorithm is $O(N \log N)$.

However, we do not consider the case, when curves intersect or share the endpoint(s). As it has been shown in [Ram99a] applying the technique of sampling leads to inadequacy of approximations in the mentioned situations. Topological inconsistencies [Ram99a] are also considered (during the process of merging). A significant advantage of this approach is the ability to refine approximations for critical areas, defined by specific practical problems. This approach makes it possible to refine a critical local region of Voronoi diagram by supporting dynamic data structures like concatenable queues [Sha75c].

Current research is implemented in software, the result is illustrated on Figures 1-2 and 5-9.

7. REFERENCES

[Prep85b] Preparata, F. P. and Shamos, M.I., Computational Geometry: An introduction. Springer-Verlag, Berlin, 1985.

[Fort87a] Fortune, S. A sweepline algorithm for Voronoi diagrams. Algorithmica, N 2, pp. 153-174, 1987.

[Sud06c] Sud, A., Govindaraju, N. and Manocha, D. Interactive computation of discrete generalized Voronoi diagrams using range culling. Proc. International Symposium on Voronoi Diagrams in Science and Engineering, P. 1-10, 2006.

[Hof99c] Hoff, K. E. , Culver, T., Keyser, J., Lin, M. and Manocha, D. Fast computation of generalized Voronoi diagrams using graphics hardware. Proc. of ACM SIGGRAPH Annual Conference on Computer Graphics, ACM, pp. 277–286, 1999.

[Ho09c] Ho, Y. J. and Liu, J. S. Collision-free curvature-bounded smooth path planning using composite bezier curve based on Voronoi diagram. IEEE International Symposium on Computational Intelligence in Robotics and Automation, Korea, pp. 463-468, 2009.

[Ary02c] Arya, S. and Malamatos, T. Linear-size approximate Voronoi diagrams, Proc. 13th ACM-SIAM Sympos. Discrete Algorithms, pp. 147–155, 2002.

[Ary02b] Arya, S., Malamatos, T. and Mount, D. M. Space-efficient approximate Voronoi diagrams, Proc. of STOC, pp. 721-730, 2002.

[Har01c] Har-Peled, S. A replacement for Voronoi diagrams of near linear size. Proc. of FOCS, pp. 94-103, 2001.

[Seo08c] Seong et al. Voronoi diagram computations for planar NURBS curves. Proc. ACM Symp. Solid & Phys. Modeling, NY, pp. 67–77, 2008.

[Ram99a] Ramamurthy, R. and Farouki, R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries: I. Theoretical foundations. J.Comput.Appl.Math. 102, pp. 119–141, 1999.

[Sha75c] Shamos, M. and Hoey, D. Closest-point problems. Proc. 16th Annu. IEEE Sympos. Found. Comput. Sci., pp. 151-162, 1975

[Ram99a] Ramamurthy, R. and Farouki, R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries: II. detailed algorithm description. J.Comput.Appl.Math. 102, pp. 253–277, 1999

[Aur13b] Aurenhammer, F., Klein, R. and Lee, D. T. Voronoi Diagrams and Delaunay Triangulations. World Scientific Publishing Co., 2013.

[Aic10a] Aichholzer, O., Aigner, W., Aurenhammer, F., Hackl, T., Jüttler, B., Pilgerstorfer, E., Rabl, M. Divide-and-conquer for Voronoi diagrams revisited". Computational Geometry: Theory and Applications, 2010, V 43, Is. 8, P. 688-699.

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

Using Images Rendered by PBRT to Train Faster R-CNN for UAV Detection

Junkai Peng[1], Changwen Zheng[2], Pin Lv[3], Tianyu Cui[4], Ye Cheng[5] and Lingyu Si[6]
Institute of Software, University of Chinese Academy of Sciences
4# South Fourth Street, Zhong Guan Cun, 100190, Beijing, P.R. China
pengjunkai15@mails.ucas.ac.cn[1], {changwen[2], lvpin[3], cuitianyu[4]}@iscas.ac.cn
chengye2010@aliyun.com[5], ls16200@my.bristol.ac.uk[6]

ABSTRACT

Deep neural networks, such as Faster R-CNN, have been widely used in object detection. However, deep neural networks usually require a large-scale dataset to achieve desirable performance. For the specific application, UAV detection, training data is extremely limited in practice. Since annotating plenty of UAV images manually can be very resource intensive and time consuming, instead, we use PBRT to render a large number of photorealistic UAV images of high variation within a reasonable time. Using PBRT ensures the realism of rendered images, which means they are indistinguishable from real photographs to some extent. Trained with our rendered images, the Faster R-CNN has an AP of 80.69% on manually annotated UAV images test set, much higher than the one only trained with COCO 2014 dataset and PASCAL VOC 2012 dataset (43.36%). Moreover, our rendered image dataset contains not only bounding boxes of all UAVs, but also locations of some important parts of UAVs and locations of all pixels covered by UAVs, which can be used for more complicated application, such as mask detection or keypoint detection.

Keywords

Object detection, deep learning, Faster R-CNN, PBRT, UAV

1 INTRODUCTION

As the rapid development of the powerful technologies of UAVs, more and more individuals can use UAVs to do creative works. Nevertheless, UAVs must be regulated in public area or no-fly zone, otherwise UAVs can become potential threats to public security and privacy.

A crucial step in the regulation of UAVs is detecting UAVs in videos rapidly. UAV detection means finding the location of each UAV for every frame of a video. More specifically, it draws a smallest rectangle that can cover all the pixels of a target UAV. Object detection in computer vision provides lots of methods to address this problem. With the development of deep learning in recent years, deeper and more complex convolutional neural networks (CNN) [1-5] have been designed to detect objects in videos efficiently, and have reached state-of-the-art performance.

However, training these CNNs requires a huge amount of data. Fortunately, several large-scale datasets are publicly available on the Internet, such as PASCAL VOC dataset [6] and COCO dataset [7], whose images are collected from the Internet and annotated through crowdsourcing. In addition, COCO dataset [7] includes not only objects' categories and positions, but also each individual's mask.

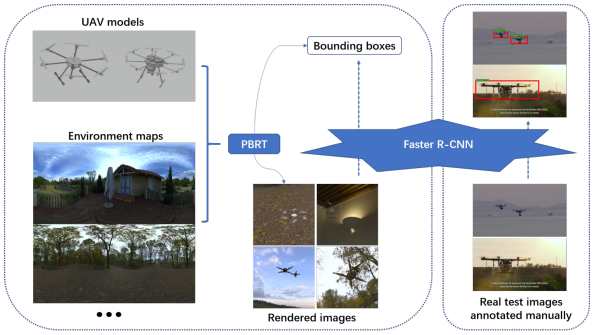


Figure 1: System overview. In contrast to previous methods, we use PBRT with environment maps as light source to render more photorealistic UAV images in a reasonable time.

Although these datasets contain a number of airplane (or aeroplane) images, more annotated UAV images are needed to promote the accuracy of UAV detection. Undoubtedly, annotating lots of UAV images through crowdsourcing is time consuming. In this manuscript, we propose to use Physically Based Rendering Toolkit

Figure 2: Some of our rendered images for training Faster R-CNN.

(PBRT) [8] (a slightly modified version) in computer graphics to render photorealistic images, and in the meantime, to calculate corresponding accurate bounding boxes. Figure 1 shows the overall process. By choosing different (1) positions and orientations of UAVs, (2) 3D models of UAVs, (3) appearance materials of UAVs, (4) camera intrinsics and extrinsics, (5) environment maps and (6) postprocessing methods of the rendered images, a large variety of images can be rendered. The greatest advantage of acquiring UAV images by rendering is that not only accurate bounding boxes of UAVs can be easily obtained, but also the positions of any parts of UAVs (e.g. lifting rotor), even all the pixels occupied by the UAVs, can be recorded. Actually, supervised learning assumes that training data and testing data should be independent and identically distributed, which means the rendered images for training must be photorealistic and diverse. This is the reason that we use PBRT to render UAV images. Figure 2 shows some of the rendered images, which are real enough and various.

Faster R-CNN trained with PASCAL VOC 2012 dataset mixed with our rendered images has an AP of 80.69% on manually annotated UAV images test set, while Faster R-CNN purely trained with PASCAL VOC 2012 dataset, which contains hundreds of airplane images, has an AP of 43.36% on the same test set. This sufficiently validate our rendered UAV images and our work lays a foundation for using images rendered by PBRT to train Faster R-CNN. Obviously, these rendered images can also be used in broader applications, for example, instance segmentation (mask R-CNN [9]) and keypoint detection.

2 RELATED WORK

Object detection. In this manuscript, we pay more attention to validating our rendered training data and talk about its possible usages in more complicated applications. Therefore, we directly use a newly released framework Detectron [10], which trained and tested a lot of state-of-the-art object detection models for our reference, to build our UAV detection system without modification.

From Fast R-CNN [1] to Faster R-CNN [2], from YOLOv1 [3] to YOLOv2 [4], more and more effective methods have been emerging constantly to make object detection faster and more accurate. Deeper and deeper CNNs need more training data to avoid overfitting. So, next we introduce some previous methods about how to use rendered images to provide extensive training data.

Using synthetically generated images. Large-scale data is of significant importance for training deep neural networks. A variety of datasets with different characteristics have promoted many fields in computer vision. For example, ImageNet dataset [11] is necessary for the breakthroughs in both object classification and detection; COCO dataset [7] plays an important role in scene understanding.

As for synthetically generated images, Dosovitskiy et al. [12] created a simple synthetic 2D dataset of flying chairs for training their network which was proved to be sufficient to predict accurate optical flow in general videos. Inspired by this, Mayer et al. [13] used a customed version of the open source 3D creation suite Blender to render three dataset for training and evaluating scene flow methods. Su et al. [14] proposed

ISBN 978-80-86943-41-1

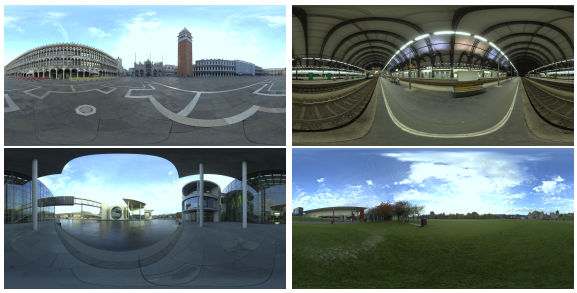


Figure 3: Some environment maps used in rendering.

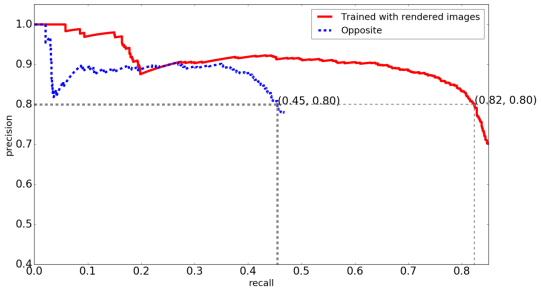


Figure 4: The precision-recall curves.

Figure 5: Left pair: using red-specular-plastic. Right pair: using aluminium. For each pair, the left image is rendered with default method and the right image is rendered with fitted NPF model and using its fitted D factor for importance sampling. Pay more attention to the white noise on the UAVs

a synthesis pipeline that generated millions of images with accurate viewpoint labels for viewpoint estimation. But rather than pursuing realistic effect, they put more effort to generate images of high diversity. Therefore they used alpha-composition to blend a rendered 3D model as foreground and a scene image as background. In contrast to them, we used environment maps as light sources to make rendered images more realistic. Peng et al. [15] also used synthetic images to train deep object detectors. They explored the complex invariance encoded in the features learned by CNN. One of their major conclusions was, when learning a detection model for a new category with no or limited labeled real data available, it was advantageous to simulate texture, color and pose in the synthetic data. Using PBRT, we can take all these factors into account easily. Aker et al. [16] simply combined background-subtracted real images to create an extensive artificial UAV dataset for training a UAV detection network. This method is only suitable for small UAVs because it does not take lighting condition into account. Cutting a medium or large UAV into another image makes the UAV look abrupt. Moreover, they need to segment some UAVs from background. In contrast to this method, we use PBRT [8] to render photorealistic UAV images, which can not only use detailed 3D models and modify their appearance materials, but also set the position of the UAV relative to the camera arbitrarily. PBRT [8] is a modern photorealistic rendering system that can even render vivid natural scenes, although it may take a considerably long time. In next section we introduce how to use PBRT with environment maps and measured materials to render real enough UAV images within a reasonable time.

3 REALISTIC IMAGE SYNTHESIS

Modeling all the objects, including UAVs, trees, houses, roads and so on, to form a natural scene will take PBRT or other renderers an unbearably long time to render an image. Previous methods simply combined real background images with rendered 3D models, which lost realism to some extent. Here we propose to use environment maps, which are images of the distant environment surrounding the rendered object. As light sources in the scene, environment maps provide illuminations shining on the UAV from all 360° angles. Usually, an environment map is synthesized by photos of a same scene taken under several specific angles. They can also be rendered by PBRT through careful design. All the 90 environment maps used to render our training set are downloaded from HDRI Haven[1]. Besides in Figure 1, more environment maps are shown in Figure 3.

Detailed UAV 3D models are another key part in rendering photorealistic images. In this manuscript we totally only use five detailed UAV models, two of which are shown in the top left corner of Figure 1. Peng et al. [15] showed a significant boost from adding more shape variation to the training data for Fast R-CNN. Therefore, it is convinced that using more 3D models to render more UAV images for training can further improve the performance of UAV detection. Unfortunately, there is still a lack of freely available and detailed UAV 3D models.

In addition, in order to increase the diversities of the training set, the MERL database [17], which includes 100 different accurately measured materials, are used in rendering. However, directly using these measured data with default importance sampling method of PBRT requires a large number of samples per pixel for rendering, which takes PBRT a relatively long time to render an image. Therefore, a BRDF model named non-parametric factor microfacet model (a modified version), which was first designed by Bagher et al. [18] and was much more accurate than other microfacet

[1] https://hdrihaven.com/

ISBN 978-80-86943-41-1

Faster R-CNN	AP
Pretrained with COCO 2014 dataset	43.03%
Finetuned with only PASCAL VOC 2012 dataset	43.36%
Finetuned with only our rendered training set (without occluded images)	56.28%
Finetuned with PASCAL VOC 2012 dataset and our rendered training set (without occluded images)	79.51%
Finetuned with PASCAL VOC 2012 dataset and our rendered training set (with occluded images)	**80.69%**

Table 1: Testing results.

Figure 6: Some of our test images. UAVs in these images can not be detected by the network only trained with COCO dataset and PASCAL VOC dataset but can be detected by the network trained with them mixed with our rendered images.

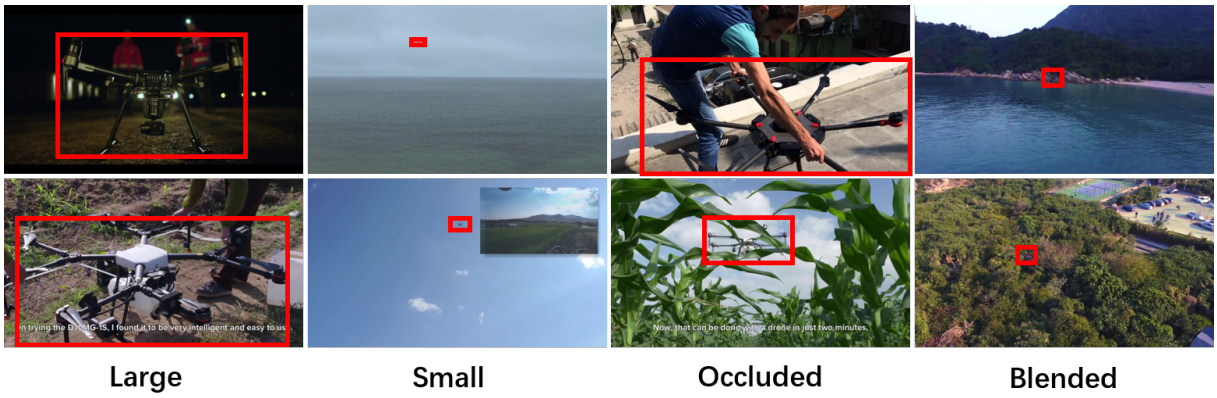


Figure 7: Some of our test images. UAVs in these images can not be detected by both the networks trained and not trained with our rendered images. It is worth noting that not all large, small, occluded or blended UAVs can not be detected.

models for fitting, is used to fit these measured data and the corresponding fitted D factors are used for importance sampling. This method dramatically reduces the needed number of samples per pixel for rendering. In other words, within the same time, this method can render a higher quality image. For example, Figure 5 shows two pairs of UAV images rendered with measured data, red-specular-plastic and aluminium respectively. For each pair of images, the left one is rendered with default importance sampling method while the right one is rendered with aforementioned method. All the images are rendered with 64 samples per pixel. Obviously, for each pair, the right image has a higher quality, especially the aluminium pair (pay more attention to the white noise on the UAVs).

Moreover, the positions and orientations of camera and UAV model can also be used to increase the diversities of the training set. Rotating around the UAV, the camera takes several photos for every 60°. The UAV also rolls $[-45°, 45°]$, for which the interval is set to 15°. The distance between the UAV and the camera is set to 4m, 8m or 12m. The field of view of the camera is about 30°. This parameter setting is not immutable.

Motion blur is not present in UAV videos except for the rotor wings. But those static UAVs should also be detected. Therefore, both static UAVs and lifting UAVs are rendered. We use Blender to split the UAV model into several parts and set rotation speed of its rotor wings to be 50 revolutions per second in PBRT scene files. The shutter speed is about 0.005 second.

Totally, we rendered 60480 UAVs images for training. It took Intel i7-4710MQ approximately 3 seconds to render an image and record its corresponding bounding box. Consequently, it took about two day to render the whole training set. Compared with annotating manually, this time is negligible. The quality and efficiency

ISBN 978-80-86943-41-1

of rendering can be further improved by using the OptiX API with the AI-accelerated denoiser [19].

4 FASTER R-CNN TRAINING AND TESTING RESULT

We use Detectron [10] provided by Facebook AI Research to finetune Faster R-CNN with our rendered training set. Detectron model zoo provides some models pretrained with COCO dataset, which can be used to initialize our network. Finally ResNet-101 model is selected as the backbone model, which has closer performance to ResNeXt-101-32x8d model but much less inference time [10]. Initialized by this pretrained model, we first trained Faster R-CNN with PASCAL VOC 2012 dataset. Then PASCAL VOC 2012 dataset is mixed with our rendered dataset to train another Faster R-CNN. The net gain from our rendered dataset is checked by comparing the performance of these two trained network.

We do not split the rendered images to evaluate the trained detection networks. Instead, many key frames cut from 10 real UAV introduction videos that are downloaded from Internet are annotated manually by us to form a test set (about 994 images including UAVs). Some of these test images are shown in Figure 6 and Figure 7. This test set is used to evaluate the trained detection network. All the testing results are shown in Table 1 and their corresponding precision-recall curves are shown in Figure 4. Actually, COCO 2014 dataset and PASCAL VOC 2012 dataset do not have any UAV images, although they have plenty of airplane (or aeroplane) images. However, Faster R-CNNs trained with them still have an AP of 43% on our test set. As shown in Figure 4, they have relatively low recall. That is, they can detect UAVs in some simple scenes but do not work for relatively complex scenes. In addition, Table 1 also shows that, finetuned with PASCAL VOC 2012 dataset, the performance of Faster R-CNN is neither increased nor decrease.

Before our rendered images are mixed into PASCAL VOC 2012 dataset, they are divided into two sets, first of which contains 38880 images that only have one UAV. The second set contains 21600 images that have two UAVs and most of them are occluded by each other. Some of these rendered images are shown in Figure 2. It is noted that Faster R-CNN trained with PASCAL VOC 2012 dataset mixed with the first set has an AP of 79.51% (Table 1), which is much higher than that only trained with PASCAL VOC 2012 dataset (43.36%). However, Faster R-CNN trained with PASCAL VOC 2012 dataset mixed with both two sets has an AP of 80.69%, which only slightly larger than 79.51%. It seems that UAVs occluded with each other in the second set do not provide more information. But in fact, there are only a few test images containing UAV occluded by people or plants (see Figure 7). In addition, Faster R-CNN purely trained with the first set has an AP of 56.28%. To some extend, the diversities of our rendered images are limited by the number of UAV models and environment maps used in rendering. Using more models, environment maps or even camera settings will definitely further promote the performance of Faster R-CNN.

Figure 8: The UAV is occluded in both images but the right one can be detected.

Figure 6 shows some images from the test set, for which the network trained with our rendered images can detect the UAVs inside them but the other one not trained with our rendered images cannot. The threshold is set to 0.7. These images span from small UAVs to large UAVs. One of the UAV is even partly occluded. Therefore, the qualitative and quantitative results show that our rendered trained images make the trained Faster R-CNN more general.

There are still some images of complex scenes that the networks neither trained nor not trained with our rendered images can detect UAVs inside them. Some of these images are shown in Figure 7. It is worth noting that not all too large, too small, occluded or blended UAVs can not be detected. As shown in Figure 8, the UAV is occluded in both images but the UAV in the right image can be detected.

These undetected images also guide the direction for us. Firstly, we need to prepare more training images which include big and detailed UAVs, or small and blur UAVs. Next, we needed to add special lighting conditions in some images. Additionally, images of UAVs occluded by persons or plants are not easy to render but images of UAVs occluded by cars, boats or houses are relatively easy. Last but not least, as indicated by Su et al. [14], using more 3D models will also help resist overfitting of the R-CNN.

5 FUTURE WORK

Besides aforementioned work, we also try to use our rendered images to train mask-RCNN and do key point detection.

Moreover, it is completely possible to render videos of flying UAVs. But if the UAVs need to interact with other objects, the scene files will be more complex and the rendering time will be longer. But with carefully designed scene files and accelerated by the OptiX API

ISBN 978-80-86943-41-1

[20], some relatively simple natural scenes are possible to be rendered in a reasonable time.

6 CONCLUSION

We present a synthetic UAV dataset with sufficient realism, variation and size. It is rendered by PBRT with measured reflection materials and environment maps that reduce the rendering time but still promise the realism. By comparing the AP of Faster R-CNN detection networks trained with and without our rendered images, we conclude that, UAV images rendered by the method mentioned above do promote the performance of the UAV detection network.

7 REFERENCES

[1] Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV) pp. 1440–1448 (2015)

[2] Ren, S.Q, He, K.M, Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(6), 1137–1149 (2017)

[3] Redmon, J. and Divvala, S. and Girshick, R. and Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 779–788 (2016)

[4] Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger. Towards Real-Time Object Detection with Region Proposal Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6517–6525 (2017)

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. In: Computer Vision – ECCV 2016 pp. 21–37 (2016)

[6] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88(2), 303–338 (2010)

[7] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Computer Vision – ECCV 2014 pp. 740–755 (2014)

[8] Pharr, M., Jakob, W., Humphreys, G.: Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann, San Francisco, 2016

[9] He, K.M., Gkioxari, G., DollÃ¡r, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV) pp. 2980-2988 (2017)

[10] Ross, G., Ilija, R., Georgia, G., Piotr, D., He, K.M.: Detectron. `https://github.com/facebookresearch/detectron`, 2018

[11] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 248-255 (2009)

[12] Dosovitskiy, A., Fischery, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Smagt, P.V.D., Cremers, D., Brox, T.: FlowNet: Learning Optical Flow with Convolutional Networks. In: 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2758-2766 (2015)

[13] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4040-4048 (2016)

[14] Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In: 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2686-2694 (2015)

[15] Peng, X.C., Sun, B.C., Ali, K, Saenko, K.: Exploring Invariances in Deep Convolutional Neural Networks Using Synthetic Images. CoRR abs/1412.7122 (2014)

[16] Aker, C., Kalkan, S.: Using deep networks for drone detection. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) pp. 1-6 (2017)

[17] Matusik, W., Pfister, H., Brand, M., McMillan, L.: A Data-Driven Reflectance Model. ACM Transactions on Graphics 22(3), 759-769 (2003)

[18] Bagher, M.M., Snyder, J., Nowrouzezahrai, D.: A Non-Parametric Factor Microfacet Model for Isotropic BRDFs. ACM Transactions on Graphics 35(5), 159:1–159:16 (2016)

[19] Chaitanya, C.R.A., Kaplanyan, A.S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., Aila, T.: Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. ACM Transactions on Graphics 36(4), 98:1–98:12 (2017)

[20] Parker, S.G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M.: OptiX: A General Purpose Ray Tracing Engine. ACM Transactions on Graphics 29(4), 66:1–66:13 (2010)

 ISBN 978-80-86943-41-1

Real-time visualization of Dynamic Unlimited Objects Instancing

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 propose a novel approach to an efficient rendering of an unlimited number of dynamic and unique 3D objects in real-time. We present an extension to the Holistic Unlimited Object Instancing (UOI) rendering pipeline and the holistic computer graphics paradigm. We called this extension Dynamic Unlimited Object Instancing rendering pipeline. Using Signed Distance Functions (SDF) for the virtual scene representation and the Holistic Scene Dynamics Function, we can control and render an unlimited number of dynamic 3D objects in real-time. In order to solve some issues of the original UOI rendering pipeline, we developed two extensions: first, a collection of holistic Dynamic operators, and, second, the Multipass Depth-Based Ray Marching rendering pipeline. The operators are used to apply affine transformations to an unlimited number of 3D objects and also to animate their materials and other attributes. In order to solve the problem of the uniform object distribution within the scene, we redefined the original definition of the scene SDF component. The virtual scene equation is divided into independent SDF components, which are rendered separately using the Multipass Depth-Based Ray Marching pipeline. Thanks to both extensions, the new version of the Holistic UOI rendering pipeline can handle 3D objects intersections what significantly enhances the realism of SDF scenes. The presented extensions to the UOI rendering pipeline are fully compatible with the Holistic UOI rendering pipeline, SDF and Sparse Voxel Octree (SVO) based algorithms. The only hardware requirement for our approach is the support for multipass rendering with compute shaders or any GPGPU API.

Keywords

Computer graphics, signed distance function, holistic programming paradigm, voxel rendering, sparse voxel octree, instancing, data-based amplification, procedural generation, fractal noises, level of detail

1 INTRODUCTION

Virtual scene geometrical complexity is one of the most common indicators used to evaluate the quality of real-time realistic image synthesis. In order to achieve the desired depth and realism of virtual worlds, the scenes should be composed of high-resolution 3D objects with detailed geometries and materials. Moreover, to provide an appropriate level of immersion of the user in the virtual environment, we have to use suitable efficient rendering techniques and algorithms to process and visualize the scenes in real time.

Over the years, many algorithms for virtual scenes management [Greene95], level of detail control [Lueb02], objects culling [Bittner04], and geometry instancing [Carucci05] have been developed. However, despite the constantly increasing computational power and memory capacity of today's GPUs, still, the main limitation of video game engines is the object space computation complexity [Jab17]. Also, one of the often overlooked factors influencing the quality of synthesized scenes, when regarded from the standpoint of the user's immersive perception of the virtual environment, is the "evolving" complexity of the virtual world that undergoes structural changes over time. We decided to focus mainly on this issue in this paper.

On the other hand, much effort has been devoted to studying alternative representations of geometry for real-time graphics. Signed Distance Functions (SDF), which derive from fractal theory and ray-tracing of quaternion Julia sets [Hart89], have found application in modern video game engines, among others for shadow map generation [Wright15] and font

rendering [Green07]. Voxel-based representations, which had been used mainly in offline computer graphics, thanks to developed Sparse Voxel Octree algorithms [Crassin11, Jab16, Domaradzki16] can now be used successfully in real-time graphics. The research we conducted on these two approaches to representing geometry for computer graphics has resulted in the development of the Holistic Unlimited Object Instancing (UOI) rendering pipeline.

The SDF-based representation has been successfully integrated with Sparse Voxel Octree algorithms in the Holistic UOI rendering pipeline [Jab17]. Thanks to the screen-space computation complexity of the SDF and SVO processing algorithms along with the newly proposed *Holistic Graphics Programming* paradigm, this allowed for a significant increase in the complexity of virtual scenes that can be rendered in real-time. By using SDFs for the scene representation integrated with SVOs as a 3D geometry representation, the idea behind object instancing was extended in that it was possible to render in real time actually an unlimited number of 3D objects created by artists.

Nevertheless, the original concept of the Holistic UOI rendering pipeline is limited in several aspects and in this paper we propose solutions to these issues. First of all, we present a novel approach to an efficient rendering of a potentially unlimited number of *dynamic* and *unique* 3D objects in real-time. By extending the original Holistic UOI rendering pipeline with *Dynamic operators* and *Holistic Scene Dynamics Functions*, we are able to add movement and animation of objects' attributes to originally static scenes. Moreover, thanks to redefining the scene SDF component, we can effectively limit the uniform object distribution artifact, which was inherent to the original definition of the component. To express with the name the functionality offered by our extension to the original UOI pipeline, we call it the *Dynamic Unlimited Object Instancing* rendering pipeline.

2 RELATED WORK

In the paper "Unlimited Object Instancing in real-time" [Jab17] the holistic computer graphics programming paradigm was introduced and embodied therein in a novel Holistic UOI rendering pipeline. Thanks to this new holistic approach to expressing scenes for computer graphics and the dedicated rendering pipeline, it was possible to process and render a potentially unlimited number of unique 3D objects in real-time. The main foundation of the presented approach was the integration of SDF and SVO algorithms in a single-pass rendering pipeline.

The original Holistic UOI rendering pipeline was based on four main components. In the context of the topic we tackle in this paper, the most important is the component of *Global operators*. It was used to control the content and complexity of the virtual scenes.

Using a collection of Global operators, it is possible to instantiate an unlimited number of 3D objects, generate and apply object variations, and control the existence of objects in the virtual scene.

Thanks to the SDF-based representation and a holistic approach to control, the memory requirements for the scene description were significantly reduced, making processing an unlimited number of the 3D object for each SDF component possible. However, the original implementation suffers from two problems, which are essential from the standpoint of realistic and immersive rendering of a virtual world.

The first issue is that the rendered worlds were static and the original architecture of the holistic pipeline makes it difficult to introduce any kind of movement to these unlimited but indeed "frozen-in-time" virtual worlds.

The second issue is related to the inherent to SDF instancing, easily noticeable artifact of the uniform distribution of objects populating the scene. In this paper, we provide the solutions to both these problems.

There is a wide selection of literature related to each component used in our Dynamic UOI rendering pipeline. The SDF-based graphics representation derives from a method introduced in the paper [Hart89] for the visualization of quaternion Julia sets. The idea of unbouding volumes presented there was then extended by Hart et al. [Hart94, Hart97] into *sphere tracing*. Given an object represented by an SDF, sphere tracing relies on iteratively traversing a ray from the eye through the projection plane towards the object. If the eye-to-object distance estimation is smaller than a predefined precision value, the ray is considered to hit the object. SDF functions can be used to create highly detailed procedural objects using SDF primitives with boolean operators. Reiner et al. [Reiner11] presented an introduction to an interactive SDF ray marching pipeline with a procedural object generation based on domain operations.

In turn, thanks to the development of SVO algorithms, the high-resolution voxel-based representation can now be used in real-time graphics applications. Due to the screen-space character of the computation complexity of the SVO rendering pipeline, numerous high-resolution 3D objects can be processed in real-time using instancing approach. Cyril Crassin was able to perform visualization of the global illumination using SVO and voxel cone tracing [Crassin11]. There are also a few promising SVO methods for object animation, deformation, and fracturing in real-time [Bau11, Wil13, Domaradzki16]. The SVO-based object representation

 ISBN 978-80-86943-41-1

has also found application in continuous LOD management [Jab16].

For these reasons, the SVO-based representation has been becoming an increasingly serious alternative to polygon-mesh representations and, as such, is a promising candidate to be utilized in the holistic rendering pipeline.

There are also a few interesting papers about procedural generation of infinite cities which are worth mentioning [Greu03, Stein14, Steninb14].

The last group of papers is related to the topic of the procedural generation of geometry by means of the data amplification approach. Since there is a vast literature on fractals and procedural graphics, below we will focus only on the papers most relevant to our work.

Ken Perlin introduced a relatively simple and efficient method for generating a space continuous, pseudo-random noise for computer graphics [Perlin02]. It has been used across the computer graphics applications from terrain generation, objects randomization to special effects. There are also many improvements to Perlin's original idea that can be relatively easily implemented in today's GPUs [Li15].

Deussen et al. [Deussen98] presented a great example of how to exploit the data based amplification approach with geometry instancing in order to create realistic plant ecosystems in non-real-time graphics engines. Due to the limited capacity of GPU memory, real-time procedural content generation is required for creating complex and unique virtual scenes.

3 HOLISTIC UNLIMITED OBJECT INSTANCING

In this section, we present a short summary of the Holistic UOI rendering pipeline which was introduced in [Jab17]. We describe the main idea behind the holistic virtual scene definition, the available features, and the architecture of the Holistic UOI rendering pipeline. In particular, we focus on the issues of the UOI rendering pipeline which we deal with in this paper.

3.1 Holistic UOI rendering pipeline

The main foundation of the Holistic UOI rendering pipeline is the integration of the SDF and SVO algorithms in a single-pass ray marching visualization pipeline [Jab17]. The holistic approach is applied to the virtual scene definition. Rather than representing a virtual scene as a collection of individual objects, the whole scene is perceived and processed in its entirety as a complex object whose geometry is described by a single and (usually) relatively simple equation.

By using this new approach to the scene representation and visualization, which was termed as the *Holistic Graphics Programming*, it is possible to process in real-time as many unique instances of 3D objects as we want. The usage of SDFs allows memory requirements for the scene description to be significantly reduced, making it possible to deal with complex and even unlimited scenes with a low memory capacity. Moreover, thanks to incorporating the SVO representation into the SDF scene description, it is possible to render high-resolution 3D objects created by artists with the usage of e.g. Physically Based Rendering materials [Pharr17].

The features of the UOI rendering pipeline are as follows:

- Real-time processing and rendering of an unlimited number of unique 3D objects in the virtual scene.
- The possibility of visualizing 3D objects created by artists.
- Compatibility with other SDF and SVO based algorithms.
- Holistic content and complexity control with a data amplification method.
- A continuous LOD management of the virtual scene.

3.2 Holistic UOI architecture

Fig. 1 presents the four components the Holistic UOI rendering pipeline.

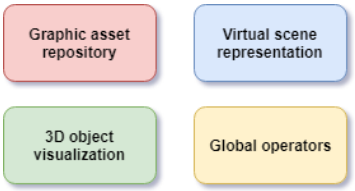


Figure 1: The components of the Holistic UOI rendering pipeline.

In this paper, we mainly focus on developing an extension to the *Global operators* component. The original paper [Jab17] introduced *Transition operators* to apply affine transformations to 3D objects. However, the capabilities of the operators were limited and they didn't take into account the passage of time in the virtual world. In the next section, we discuss two major issues of the original concept of the UOI rendering pipeline.

3.3 Holistic UOI issues

The holistic UOI rendering pipeline offers the possibility to handle an unlimited number of unique 3D objects in the virtual scene in real-time.

The original *Global operators* component gathers various instancing, geometry and material operators. Although there was a class of object transformation operators available, they did not offer satisfactory results. The two main issues were the uniform distribution of objects within the scene and no support for possible objects' intersections.

ISBN 978-80-86943-41-1

3.3.1 *Uniform object distribution problem*

The first problem pertains to the construction of the *Instancing operator*—the principal operator of the holistic UOI approach. In order to generate an unlimited number of 3D objects, a modulo function is applied to the scene distance function. The result is that a single scene SDF component, which represents an object, is repeated with a defined interval and, thus, a uniform grid of the object's copies is generated, populating the virtual world.

Fig. 2 presents rendering results of a virtual scene represented by a single scene SDF component with a modulo instancing operator applied.

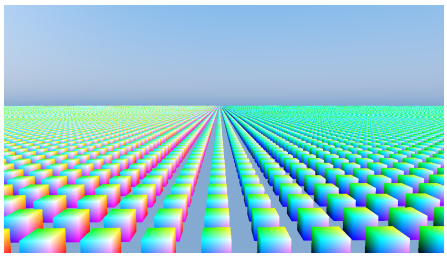


Figure 2: Uniform object distibution problem visible on single SDF component scene with Instancing Operator applied.

The original Holistic UOI rendering pipeline offers a collection of operators that can be used to reduce the visibility of this artifact—for example, the *Existence operator* could be applied to partially overcome this issue. Nevertheless, despite the application of the operator, there are always many vantage points from which the uniform object distribution is still noticeable, as we can see in Fig. 3.

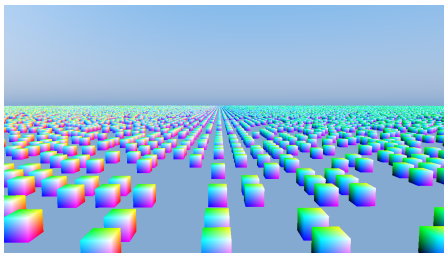


Figure 3: Uniform object distibution problem visible on single SDF component scene with *Existence operators* applied.

3.3.2 *No 3D objects intersection support*

The second problem is related to the integration of SFD and SVO in a single-pass rendering pipeline. The biggest implementation challenge for the Holistic UOI development was dealing with potential object occlusion errors [Jab17]. In the Holistic UOI rendering pipeline occlusion errors were fixed using multiple ray marching iterations. If an occlusion error occured, the grid cell coordinates and the SDF component id were stored. Then, the distance to the grid cell from the previous ray marching iteration was calculated and subtracted from the scene equation in the next iteration of ray marching algorithm. Thanks to that, the distance to the potentially occluded 3D objects could be found.

Although the algorithm is relatively simple and efficient, it also causes a serious problem, because cutting off the previous grid hit by a ray may result in that the 3D objects associated with the cell and potentially intersected by the ray are removed from the scene, too. In order to solve this issue, it is necessary to find the distance of the intersection between the multiple SDF components and apply it to the virtual scene definition [Jab17]. It means that as the result the complexity of the algorithm increases significantly.

In this paper, we propose a different solution to this problem—rather then the original single pass rendering, we make use of the Multipass Depth-Based Ray Marching (Sec. 4.3).

4 DYNAMIC UNLIMITED OBJECT INSTANCING

In this section, we describe the developed extension to the Holistic UOI rendering pipeline which we called *Dynamic Unlimited Object Instancing*. The Dynamic UOI rendering pipeline is based on the following three components:

1. **Holistic Scene Dynamics Function**—a continuous function parameterized by time and used to procedurally generate unique, dynamic variations of 3D objects populating the virtual scene.
2. **Dynamic Operators**—an extension to the original collection of the *Global operators* from the original Holistic UOI rendering pipeline. The dynamic operators are used to apply unique affine transformation and material animation to 3D objects. They utilize the *Holistic Scene Dynamics Function* to calculate dynamic variations per 3D object.
3. **Multipass Depth-Based Ray Marching**—an extension to the Holistic UOI pipeline which is based on a multipass rendering rather than—as it was in the original implementation—a single-pass rendering.

4.1 Holistic Scene Dynamics Function

The first component of the *Dynamic UOI* rendering pipeline is the *Holistic Scene Dynamics Function* (the HSD function for short). Though the function is an integral part of the *Dynamic Operators* component, we decided to define it as an independent component for the following reasons:

First, the HSD function is a perfect example of the implementation of the *Holistic Graphics Programming*

 ISBN 978-80-86943-41-1

paradigm. Instead of controlling each 3D object in the virtual scene independently, we animate the whole scene by means of a relatively simple equation. Secondly, the form of the function strongly depends on the scene content. For example, a different HSD function will be used to control an animation of flying 3D objects and a different one to sway grass under the wind.

In general, the HSD function can be expressed as:

$$f_{HSD} : D \times t \rightarrow V \tag{1}$$

where:

V = a dynamics variation for a given 3D object
D = the object input data
t = the current simulation time

For example, a HSD function implemented using Perlin's noise algorithm could generate a color and other material attributes of a 3D object as well as its transformation matrix as an output by using the object's world space position or its grid cell as an input.

A good example of a potential application of the HSD function is the impact of wind. Based on the passing time, the world space position and the SDF component unique id, we could calculate, for example, translation matrices for 3D objects.

It can be expressed as a simple pseudorandom noise generator or by using a more sophisticated method based on, for example, Fractional Brownian Motions [Mandelbrot68] or Vector Fields [Chen11]. In this paper, we use relatively simple HSD functions based on trigonometric functions and continuous noise generators.

4.2 Dynamic Operators

The second component of the Dynamic UOI rendering pipeline is a collection of *Dynamic Operators* that are used to apply changes to the static virtual scene, processed using the rendering pipeline.

Thanks to the Dynamic Operators, 3D objects can be transformed with unique affine transformations per instance and/or have their attributes animated in real-time. The Dynamic Operators extend the *Global operators* collection with the additional time dimension [Jab17].

4.2.1 Dynamic Operators architecture

The processing pipeline of the Dynamic Operators slightly differs from that related to the remaining operators. This is particularly evident in the example of the transformation operators which were applied in the original holistic approach using following processing pipeline [Jab17]:

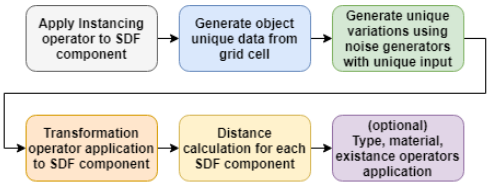


Figure 4: Global operators processing pipeline from original Holistic UOI rendering pipeline.

In the case of the Dynamic Operators, we need to perform an additional processing pass at the beginning of the holistic operator's application pipeline. In order to apply dynamic transformations for generated 3D objects, it is required to apply an additional transformation to a position on the ray from the camera to an SDF component at each iteration of the ray marching algorithm. It is required to preserve the correct form of the SDF.

The *Instancing operator* from the Holistic UOI rendering pipeline returns an object instance grid cell vector [Jab17]. The remaining *Global operators* used this value as an input for generating variations. For the Dynamic operators, we need to apply a transformation to the ray before applying *Intancing operator*. It means that we need to calculate the grid cell vector independently from the operator.

The architecture of the pipeline for the newly proposed operators takes the following form:

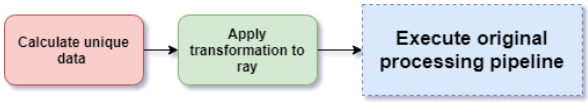


Figure 5: Global operators processing pipeline developed for Dynamic UOI rendering pipeline.

Using the SDF-based object representation, the development of additional processing paseses is relatively simple. The cube distance function, which represents base scene SDF component [Jab17], can be extended as:

$$\begin{aligned} gridCell &= floor((p + interval * 0.5)/interval) \\ variation &= f_{HSD}(time, gridCell) \\ &TrasOp(p) \\ &InstancingOp(p, interval) \\ &RotOp(p) \\ &ScaleOp(p) \\ distance &= length(max(abs(p) - size), 0) \end{aligned} \tag{2}$$

where:

$gridCell$ = an object instance grid cell
$variation$ = an object instance dynamics variation
f_{HSD} = the HSD function

ISBN 978-80-86943-41-1

$TransOp$	= Translation operator
$InstancingOp$	= Instancing operator
$RotOp$	= Rotation operator
$ScaleOp$	= Scale operator
$time$	= the elapsed simulation time
$interval$	= the repeat interval
$distance$	= the distance from the eye to the object
p	= a point on the ray from the eye to the object
$size$	= the scene SDF component cube size

4.2.2 *Dynamic Operators application results*

In this section, we present results of rendering virtual scenes with Dynamic Operators applied. In the following examples, we used a simple HSD function implemented with the use of the trigonometric functions supported by hardware.

Fig. 6 presents results of rendering a scene represented by a single scene SDF component with dynamic transformation operators applied.

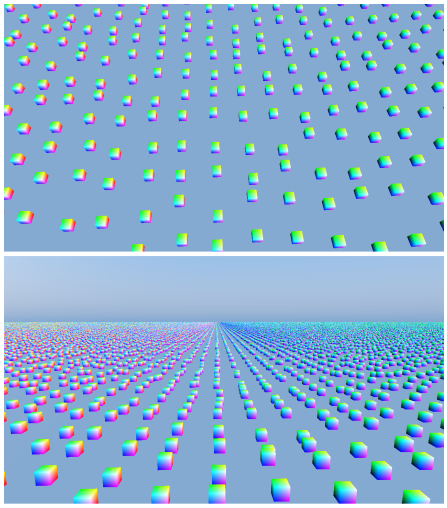


Figure 6: *Dynamic Operators* applied for the virtual scene represented by single SDF component in 2D.

The results show that the dynamic operators effectively solve the issue of the uniform object distribution. Moreover, they exemplify the possibility of the processing and visualization of an unlimited number of unique, dynamic 3D objects in real-time. They are also a good example of an implementation of the holistic programming paradigm accompanied by the data amplification approach.

Although the Dynamic operators are designed mainly as an extension to the *Transformation operators* from the original Holistic UOI, their usage is not limited only to 3D objects affine transformations. They could be also used to animate other objects attributes, e.g. the albedo color or the material's roughness values.

4.3 Multipass Depth-Based Ray Marching

The third component of the Dynamic UOI rendering pipeline is *Multipass Depth-Based Ray Marching* which turns a single-pass rendering pipeline into a multipass rendering pipeline.

The Dynamic UOI rendering pipeline with *Multipass Depth-Based Ray Marching* was developed in order to fulfill the following requirements:

- Support for 3D object intersections.
- Classic triangle rasterization rendering results integration support.
- Optimization and LOD management features for complex scenes.

4.3.1 *Scene SDF component redefinition*

In the original paper [Jab17], a virtual scene was represented using a single distance equation. Thanks to that, the whole rendering was performed in a single-pass. However, the available occlusion error-fixing algorithms do not support intersections between scene SDF components. In order to solve this issue, we decided to redefine the scene SDF function.

In order to render a scene with intersecting scene SDF components, we define each component as an independent virtual scene equation and render using a separate rendering pass. Then, so as to integrate the rendering results, a custom-made depth buffer is used.

The depth buffer is created as a second floating-point render target and utilized along with the ray-marching pipeline to store the minimum distance values obtained from the ray-marching passes. The depth buffer can be treated as another scene SDF component at the next rendering pass.

The integration of depth testing (if necessary for subsequent rendering passes) with the ray marching pipeline is quite simple. We need only to apply an additional check if the current distance traveled by the ray is smaller than the appropriate value in the depth buffer filled in in the previous passes.

4.3.2 *Multipass rendering pipeline*

A good example of the usage of the *Multipass Depth-Based Ray Marching* is rendering an open-world scene we prepared for this paper. The test scene consists of the following elements:

- **Terrain SDF component**—a procedural terrain distance function based on heightmap ray marching.
- **Trees SDF component**—objects of 3D trees created by artists with material and type operators applied. The component utilizes the terrain SDF function in order to snap 3D objects to terrain height by using *Translation operator* and *Existence operator*. Moreover, Dynamic operators are used to implement wind movement.

ISBN 978-80-86943-41-1

- **Grass SDF component**—grass objects with user-defined textures. Type and Material operators applied. 3D objects are snapped to the terrain with Translation and Existence operators applied in the same way as the previous component. Dynamic operators are also used to apply wind movement.

Fig. 7 presents outcomes of the subsequent passes of rendering the test scene. A more detailed discussion on the performance results of *Multipass Depth-Based Ray Marching* is given in the next section.

5 RENDERING AND PERFORMANCE RESULTS

All the given timings were obtained on Intel Core i5-8600K CPU with Nvidia GeForce GTX 1060 GPU and the algorithms were implemented using OpenGL 4.6 API with C++17 for Windows 10 64-bit.

We utilized 3D models Stanford Repository models [Stanford11] and other public resources [CGTrader, Sinnaeve] as test objects. In the tested scenes, we used the SDF function based on the online articles by Inigo Quilez [Iniqo08] and Alexander Alekseev [Aleksaeev14].

We prepared three virtual scenes: *Stanford*, *Terrain* and *Ocean*. For the second and third ones, we were using *Multipass Depth-Based Ray Marching* to handle 3D objects intersections and *Dynamic operators* to add movement to our scenes. The content of the *Terrain* scene was described in Sec. 4.3. The *Ocean* scene contains one scene SDF component for a procedural ocean and a second one for a herd of balloons. All the scenes are using the vast collection of *Global operators*, including instancing, type, material, and existence operators along with the newly developed *Dynamic operators* described in Sec. 4.2.

All the 3D objects are represented by SVOs with 10 levels of detail (1024 x 1024 x 1024 voxelization). Each voxel stored a compressed normal vector and texture coordinates.

The obtained rendering times (given in the figures) prove that the developed rendering pipeline is efficient and offers real-time performance. Moreover, the presented images show that the application of the Dynamic UOI rendering pipeline makes it possible to limit the noticeable regularity in object distribution inherent to the original algorithm.

The use of *Multipass Depth-Based Ray Marching* allows for handling 3D object intersections what effectively increases the depth and realism of rendered scenes. It also solves the limitation of the original occlusion error-fixing algorithm. Finally, introducing the newly developed *Holistic Scene Dynamics Function* component extends the holistic programming paradigm with movement and other possible changes to originally static objects.

As a part of the rendering performance tests, we performed an additional comparison test between the compute-shader-based and the pixel-shader-based rendering pipelines. In all performed test the rendering pipeline based on compute shaders was operating much faster. In our opinion, the compute-shader-based variant which offers the full control over the shader invocations is a better choice for a Holistic UOI implementation. Nevertheless, one should be aware that, unlike as in the case of the traditional triangle rasterization pipeline, the compute-shader implementation requires one to pay very close attention to every single line of code and even the number of registers in use.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel approach to efficient rendering of an unlimited number of dynamic and unique 3D objects in real-time. Thanks to the developed extensions to the *Holistic Unlimited Object Instancing* based on the *Holistic Graphics Programming* paradigm, we successfully limited issues featuring the original approach.

Using the developed *Dynamic Operators* along with the *Holistic Scene Dynamics Function*, we can limit the artifact of the 3D object uniform distribution. Moreover, the introduction of changes in position and other attributes of the 3D objects populating the scene significantly increase the depth and the level of immersion featuring the virtual words created and rendered with the holistic approach.

Another issue of the original method—no support for 3D object intersections we solved using *Multipass Depth-Based Ray Marching*. The redefinition of the scene SDF component allowed for authoring complex virtual scenes by means of an efficient and relatively simple method. What's more, the implementation of the multipass rendering pipeline made the integration of SDF-based objects in the virtual scene much simpler. Moreover, thanks to incorporating the depth buffer into the ray-marching rendering, a depth-based integration with results obtained with the standard triangle-rasterization pipeline is possible (e.g., for particle effects, skeletal animation or animated, user-controlled 3D objects).

One should also note that *Multipass Depth-Based Ray Marching* makes it possible to define additional rendering optimization features. For example, each rendering pass could use different ray marching parameters (ray iteration number, precision, near/far planes, etc.) or even a different render target resolution.

An obvious step forward is an implementation of a more advanced *Holistic Scene Dynamics Function*.

 ISBN 978-80-86943-41-1

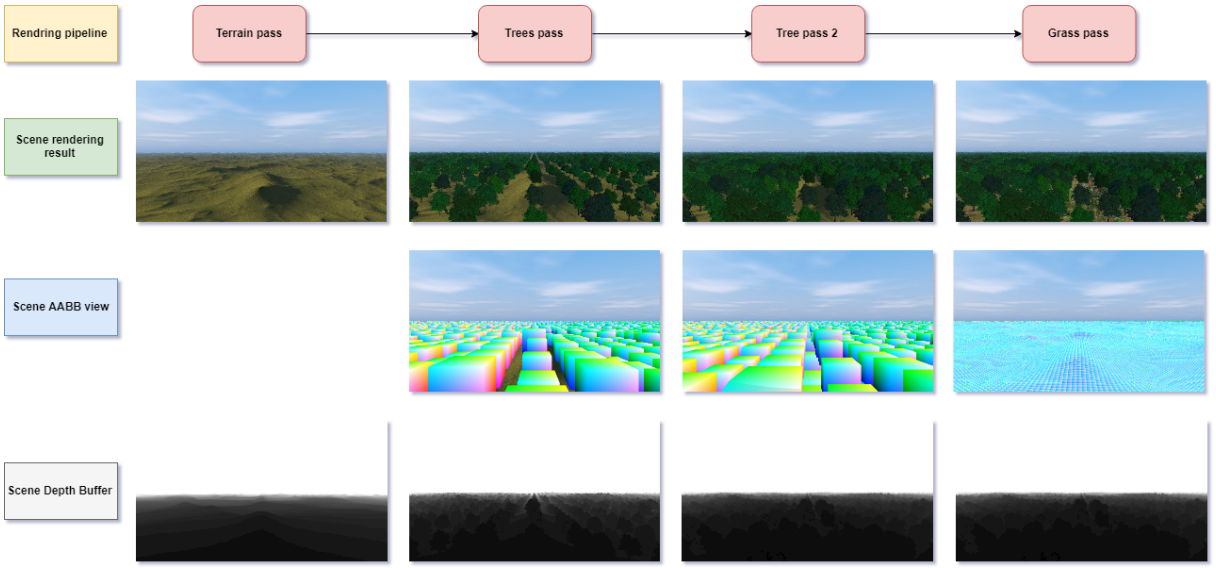
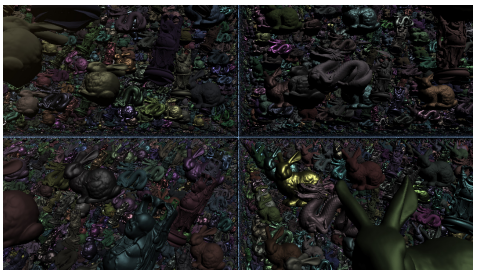
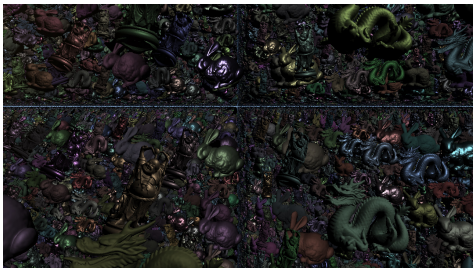


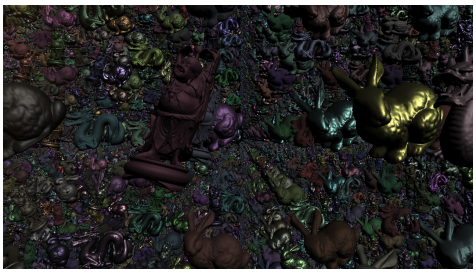
Figure 7: Multipass Depth-Based Ray Marching rendering pipeline application for test scene with multiple scene SDF components.



(a)

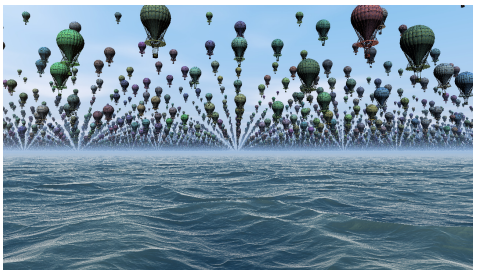


(b)



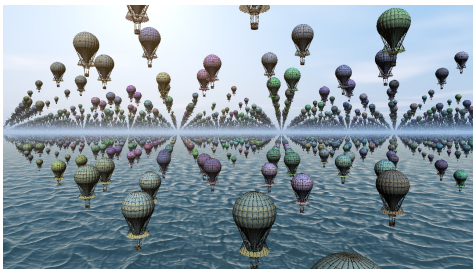
(c)

Figure 8: *Stanford* virtual scene with Dynamic operators applied with collection of original UOI operators. 40-50 FPS on Nvidia GeForce GTX 1060.



(a)

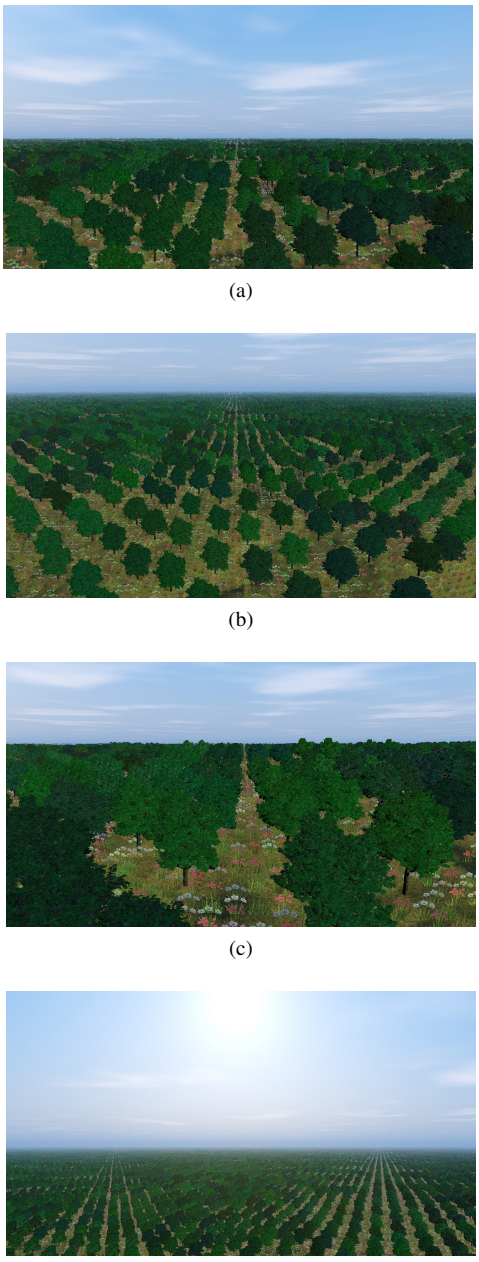
(b)



(c)

Figure 9: *Ocean* virtual scene with Multipass Depth-Based Ray Marching and Dynamic operators applied. 50-60 FPS on Nvidia GeForce GTX 1060.

ISBN 978-80-86943-41-1



(a)

(b)

(c)

(d)

Figure 10: *Terrain* virtual scene with Multipass Depth-Based Ray Marching and Dynamic operators applied. 50-60 FPS on Nvidia GeForce GTX 1060.

A good idea seems to be the usage of a procedurally generated model of wind for realistic influencing 3D objects such as balloons, grass, etc. In our opinion, a collection of specialized movement functions should become a next important component of the Holistic UOI rendering pipeline. Also, a further optimization and extension to the *Dynamic Operators* should be taken into account in future work.

A further optimization is also possible for the *Multipass Depth-based Ray Marching*. For example, the integration with the Hierarchical Z-buffer [Greene93] algorithm instead of simple depth testing seems to be easy to implement. We suspect that it could result in a significant processing performance increase.

Finally, increasing the complexity of virtual scenes which is now possible by using dynamic objects in multiple rendering passes requires an additional research concerning the level of detail management for such a rendering pipeline.

7 REFERENCES

[Aleksaeev14] Seascape, Alekseev, A., https://www.shadertoy.com/user/TDM

[Bau11] Bautembach, D., Animated sparse voxel octrees, Bachelor Thesis, University of Hamburg, 2011.

[Bittner04] Bittner, J., M. Wimmer, H. Piringer, W. Purgathofer, Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful, Computer Graphics Forum, vol. 23 no. 3, pp. 615-624, 2004.

[Carucci05] Carucci, F., Inside Geometry Instancing, in Matt Pharr, ed., GPU Gems 2, Addison-Wesley, pp. 47-67, 2005.

[CGTrader] CG Trader, 3D models for VR / AR, 3D printing and computer graphics, http://www.cgtrader.com.

[Chen11] Chen, Cheng-Kai and Yan, Shi and Yu, Hongfeng and Max, Nelson and Ma, Kwan-Liu, An Illustrative Visualization Framework for 3D Vector Fields, Comput. Graph. Forum, number 7, vol. 30, pages 1941-1951, 2011.

[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.

[Deussen98] Deussen, O., Hanrahan, P., Lintermann, B., Mesh, R., Pharr, M., Prusinkiewicz, P., Realistic modeling and rendering of plant ecosystems, Proceedings of SIGGRAPH 98, Orlando, Florida, July 19-24, 1998, In Computer Graphics Proceedings, Annual Conference Series, 1998, ACM SIGGRAPH, pages 275-286.

[Domaradzki16] Domaradzki, J., Martyn, T., Fracturing Sparse-Voxel-Octree objects using dynamical Voronoi patterns, Computer Graphics, Visualization and Computer Vision WSCG 2016. Full Papers Proceedings / Pan Zhigeng, Skala Vaclav (red.), Computer Science Research Notes, vol. 2601, 2016, Vaclav Skala - UNION Agency, ISBN 978-80-86943-57-2, pages 37-46.

[Greene93] Greene, Ned and Kass, Michael and Miller, Gavin S. P., Hierarchical Z-buffer visibility, SIGGRAPH In Proceedings, 1993.

ISBN 978-80-86943-41-1

[Greene95] Greene, N., Hierarchical Rendering of Complex Environments, Ph.D. Thesis, University of California at Santa Cruz, Report no. UCSC-CRL-95-27, June 1995.

[Green07] Green, C., Improved alpha-tested magnification for vector textures and special effects, Proceeding SIGGRAPH '07 ACM SIGGRAPH 2007 courses, pages 9-18.

[Greu03] Greutner, S., Parker, J., Stewart, N., and Leach, G., Real-time procedural generation of 'pseudo infinite' cities. In Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE '03). ACM, New York, NY, USA, 87-ff. DOI=http://dx.doi.org/10.1145/604471.604490

[Hart89] Hart, J., C., Sandin, D., J., Kaufmann, L., H., Ray Tracing Deterministic 3-D Fractals Computer Graphics 23(3), (Proc. SIGGRAPH 89,) July 1989, pages 289-296.

[Hart94] Hart, J., C., Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces, The Visual Computer, Volume 12, pages 527-545.

[Hart97] Hart, J., C., Implicit Representations of Rough Surfaces Computer Graphics forum, Volume 16, Issue 2, June 1997, pages 91-99

[Iniqo08] Quilez, I., Modeling with distance functions, http://iquilezles.org/, 1994-2017.

[Jab16] Jabłoński, Sz., Martyn, T., Real-Time Rendering of Continuous Levels of Detail for Sparse Voxel Octrees, Computer Graphics, Visualization and Computer Vision WSCG 2016. Short Papers Proceedings / Skala Vaclav (red.), Computer Science Research Notes, vol. 2602, 2016, Vaclav Skala - UNION Agency, ISBN 978-80-86943-58-9, pages 79-88.

[Jab17] Jabłoński, Sz., Martyn, T., Unlimited Object Instancing in real time, Computer Graphics, Visualization and Computer Vision WSCG 2017. Short Papers Proceedings / Skala Vaclav (red.), Computer Science Research Notes, vol. 2702, 2017, Vaclav Skala - UNION Agency, ISBN 978-80-86943-50-3, pages 91-100.

[Li15] Li, H., Tou, X., Liu, Y., Jiang, X., A Parallel Algorithm Using Perlin Noise Superposition Method for Terrain Generation Based on CUDA architecture, International Conference on Materials Engineering and Information Technology Applications, 2015.

[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.

[Mandelbrot68] Mandelbrot, B. B. and van Ness, J. W., Fractional Brownian motions, fractional noises and applications, SIAM Review, vol. 10, pages 422-437, 1968.

[Perlin02] Perlin, K., Improving Noise, ACM Transactions on Graphics, vol. 21, pages 681-682, 2002.

[Pharr17] Pharr, Matt and , and Jakob, Wenzel and , and Humphreys, Greg, Physically Based Rendering (Third Edition), Morgan Kaufmann, Boston 2017, ISBN 978-0-12-800645-0.

[Reiner11] Reiner, T., Mückl, G., Dachsbacher, C., Interactive modeling of implicit surfaces using a direct visualization approach with signed distance functions, Computers and Graphics, Volume 35 Issue 3, June, 2011, pages 596-603.

[Sinnaeve] A collection of free CG resources provided by Midge Sinnaeve, https://themantissa.net.

[Stanford11] The Stanford 3D Scanning Repository, Stanford University, 22 Dec 2010, Retrieved 17 July 2011.

[Stein14] Steinberger, M., Kenzel, M., Kainz, B., Mueller, J., Wonka, P., Schmalstieg, D., Parallel Generation of Architecture on the GPU, Eurographics 2014

[Steninb14] Steinberger, M., Kenzel, M., Kainz, B., Wonka, P., Schmalstieg, D., On-the-fly Generation and Rendering of Infinite Cities on the GPU, Eurographics 2014

[Wil13] Willcocks, C. G., Sparse volumetric deformation, Ph.D. dissertation, Durham University, 2013.

[Wright15] Dynamic Occlusion with Signed Distance Fields, Advances in Real-Time Rendering in Games, SIGGRAPH 2015.

Accelerating Evolutionary Construction Tree Extraction via Graph Partitioning

Markus Friedrich, Sebastian Feld, Thomy Phan
Institute for Computer Science
Ludwig-Maximilians-University Munich
Oettingenstr. 67
80538 Munich, Germany
{markus.friedrich|sebastian.feld|thomy.phan}@ifi.lmu.de

Pierre-Alain Fayolle
Division of Information and Systems
The University of Aizu
Aizu-Wakamatsu City
965-8580 Fukushima, Japan
fayolle@u-aizu.ac.jp

ABSTRACT

Extracting a Construction Tree from potentially noisy point clouds is an important aspect of Reverse Engineering tasks in Computer Aided Design. Solutions based on algorithmic geometry impose constraints on usable model representations (e.g. quadric surfaces only) and noise robustness. Re-formulating the problem as a combinatorial optimization problem and solving it with an Evolutionary Algorithm can mitigate some of these constraints at the cost of increased computational complexity. This paper proposes a graph-based search space partitioning scheme that is able to accelerate Evolutionary Construction Tree extraction while exploiting parallelization capabilities of modern CPUs. The evaluation indicates a speed-up up to a factor of 46.6 compared to the baseline approach while resulting tree sizes increased by 25.2% to 88.6%.

Keywords

3-d Reconstruction, Reverse Engineering, Computer Aided Design, Constructive Solid Geometry, Evolutionary Algorithms, Graph Theory

1 INTRODUCTION

Reverse Engineering (RE) – i.e., the recovery of a model's geometric representation from potentially noisy and incomplete sensor data – is an important aspect of modern Computer Aided Design (CAD) pipelines. It allows for convenient model editing based on real-world physical objects, thus simplifying and accelerating the product design process. An expressive and intuitive model representation scheme extensively used in solid modeling is Constructive Solid Geometry (CSG). It describes complex rigid solids by a binary tree with regularized Boolean set-operations (e.g., union, intersection, subtraction) as inner nodes and primitive solids (e.g., cubes, spheres, cylinders and cones) as leaves. Such a tree is also known as a model's Construction Tree. Due to the popularity of CSG in CAD, it is desirable to have tools at hand that are able to reliably recover a model's CSG-tree from its point cloud representation stemming from sensor recordings. CSG-tree generation might be solved by converting the input point cloud to a Boundary Representation (B-rep) and then by conversion of the B-rep to CSG with methods based on algorithmic geometry that usually require exact geometric intersection computations [SV93, BC04]. These approaches are usually restricted to a single model representation for primitives, e.g. a surface description that uses quadrics, and can be sensitive to inexact representations.

To overcome these constraints, CSG-tree generation can be formulated as a combinatorial optimization problem over the possible permutations of primitives and set-operations for a fixed maximum CSG-tree depth. Metaheuristics, like Genetic Algorithms (GAs) can then be employed for optimization [Mit98].

One of the most severe disadvantages of GA-based solutions are computation times of minutes and hours for comparably small models (less than 10 primitives) [FP16]. This issue is addressed in this paper.

The basic idea of the described acceleration scheme is to exploit spatial relationships between primitives: Primitives that do not overlap spatially are not considered to be operands of a CSG-operation. This knowledge can be used to partition overlapping primitives and to compute partial per-partition results that are later on merged into a single CSG-tree. In particular, this paper makes the following contributions:

- An acceleration scheme based on spatial search space partitioning together with a robust merge mechanism.
- A description and analysis of parallelization strategies for the proposed algorithms.

The paper has the following structure: Section 2 discusses related work in the field of CSG-tree extraction and surface reconstruction. It is followed by an introduction to the theoretical principles of the proposed method (Section 3). The problem to solve is detailed in Section 4. The proposed solution is described in Section 5 and evaluated in Section 6. Section 7 summarizes the results and sketches possible future work.

2 RELATED WORK

This work is related to different domains such as surface reconstruction from discrete point clouds, Reverse Engineering of solid models and conversion from B-rep to CSG. In this section, important related work in these domains is briefly discussed.

2.1 Surface Reconstruction

The problem of reconstructing a surface from a discrete point cloud has been the subject of much attention in computer graphics. The most popular methods include fitting implicit surfaces such as [OBA+03], or Poisson surface reconstruction [KH13], among others. The recent work of Berger et al. [BTS+17] presents a wide survey of the topic. Using these methods, the reconstructed objects lack information that can be used for inspection or re-use of the object in further modeling.

2.2 Reverse Engineering and B-rep to CSG Conversion

The goal of Reverse Engineering is the creation of consistent geometric models from point cloud data [VMC97, BMV01]. They usually output B-rep models made of parametric patches.
The conversion from B-rep to CSG was first investigated for two-dimensional, linear polygons, then later extended by Shapiro et al. for handling curved polygons [SV91b, Sha01]. The extension to three-dimensional objects was initially solved by Shapiro and Vossler in [SV91a, SV93] and later improved by Buchele and Crawford in [BC04]. These works rely on the fact that surfaces are composed of quadric surface patches. Another issue is the handling of inexact representations. These methods work under the assumption that the patches form a clean partition of the target solid. However, in practice, we are dealing with input point clouds that are potentially noisy, contain holes, or have additional details and thus the fitted primitives may not fit perfectly. This could impact the cellular classification on which these methods rely.

2.3 Point Cloud to CSG Construction

In [XF14], a greedy approach is used to build a CSG-representation with cuboids as primitives. This approach is limited to the reconstruction of buildings. Close to the proposed approach are methods that handle noisy and incomplete point clouds such as [SWK07] for fitting primitives and methods that try to convert them to a higher level representation such as [FP16], see also [BTS+17, Sections 7 and 8]. One of the goals of this work is to improve the running time of the Evolutionary Algorithm used in [FP16] via geometric consideration, i.e. the overlapping in space of primitives.

3 BACKGROUND

3.1 Point Cloud to CSG-Tree Pipeline

The extraction of a CSG-tree from a point cloud poses a complex problem which is usually solved with a processing pipeline that comprises the following steps:

1. **Point cloud generation and pre-processing:** Point clouds are generated by laser scanners or tactile measurement devices. Other techniques use photogrammetric algorithms to gather depth information from (un-)calibrated camera images [HZ03]. Measured point clouds usually contain significant amounts of noise and outliers. These can be trimmed from the data-set using e.g. statistical approaches [RC11].
2. **Point cloud segmentation and primitive fitting:** The point cloud must be segmented and primitive parameters have to be fitted to the corresponding points. Approaches that fulfill both tasks for simple geometric shapes are e.g. specialized variants of the Random Sample Consensus (RANSAC) technique [SWK07].
3. **CSG-tree generation:** CSG-tree generation can be done with methods based on algorithmic geometry such as [SV93, BC04], or via evolutionary approaches such as [FP16] for handling inexact representations.
4. **CSG-tree optimization:** The resulting CSG-tree might not be optimal in terms of size and depth. Additional optimization techniques can simplify the tree structure [SV91a].

3.2 Primitive Description

Primitives are basic shapes located at CSG-tree leaves. A primitive p is fully described by its signed distance function $f_p : \mathbb{R}^3 \mapsto \mathbb{R}$. The surface of p is implicitly defined by the zero-set of f_p: $\{x \in \mathbb{R}^3 : f_p(x) = 0\}$. Its surface normal at point $x \in \mathbb{R}^3$ is given by the gradient $\nabla f_p(x)$. If the gradient does not exist at x or is too expensive to compute, finite difference approximations can be used.

ISBN 978-80-86943-41-1

3.3 Boolean Set-Operations

The set-operations intersection, union, complement and subtraction are implemented using min- and max-functions [Ric73]:

- Intersection: $S_1 \cap S_2 := \min(f_{S_1}, f_{S_2})$
- Union: $S_1 \cup S_2 := \max(f_{S_1}, f_{S_2})$
- Complement: $\overline{S} := -f_S$
- Subtraction: $S_1 \setminus S_2 := S_1 \cap \overline{S_2}$

where S_i is the solid corresponding to the set $\{x \in \mathbb{R}^3 : f_{S_i} \geq 0\}$ $(i = 1,2)$. In the following, the considered Boolean set-operations are intersection, union, complement and subtraction.

3.4 Evolutionary Algorithms

Evolutionary Algorithms are biology-inspired, stochastic metaheuristics for solving optimization problems [ES+03].
The optimization process starts with a randomly initialized population of individual candidates sampled from the problem's search space (initialization). In each iteration, candidates are ranked according to their fitness by evaluating the so-called fitness function. The best candidates are selected to be the next generation's parents (parent selection). Parents are then recombined (crossover) and mutated (mutation) to create offspring. The new population is then filled with the offspring together with selected surviving individuals (survivor selection) from the current population. This procedure is repeated until a certain termination criteria is met (termination). See Fig. 1 for an overview.
Evolutionary Algorithms are especially useful for solving combinatorial optimization problems [ES+03].

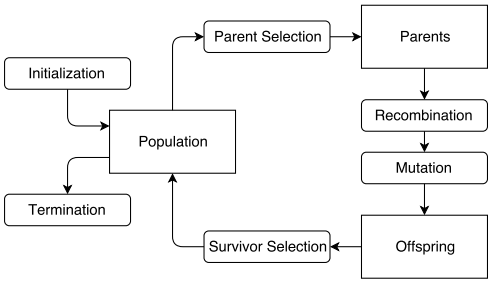


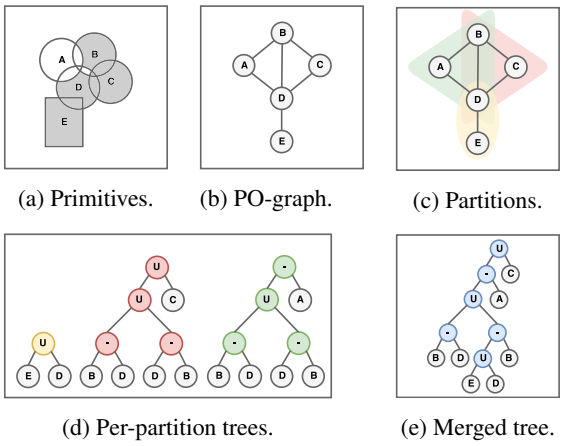
Figure 1: The optimization process described by an Evolutionary Algorithm (derived from [ES+03]).

4 PROBLEM STATEMENT

The problem of accelerating GA-based CSG-tree extraction from point clouds is considered as the open research question addressed by this paper. The focus is on CSG-tree generation and optimization (step 3 and 4 of the pipeline detailed in Section 3.1). As input, a point-set of potentially noisy 3-d measurements of a connected geometric model is considered. We also assume that the point-set is already segmented with fitted primitives, using techniques depicted in step 1 and 2 of the pipeline described in Section 3.1.
The desired output is a CSG-tree that represents the scanned real-world model as accurately as possible. A measure for accuracy is given by the distance between the CSG-tree induced surface and the points of the input point cloud. CSG-tree extraction approaches based on a GA [FP16] can handle inaccuracies but come with the disadvantage of potentially high computation times.

5 CONCEPT

The basic idea for accelerating CSG-tree extraction is to partition the search space into independent groups of spatially overlapping primitives. This exploits the fact that primitives that do not overlap are not considered to be operands of a CSG-operation. CSG-tree extraction is then conducted on a per-partition level. Finally, resulting trees are combined in a subsequent merge step without loss of result quality and correctness.
An overview of the full CSG-tree extraction pipeline is depicted in Fig. 2. Each of the steps is described in detail in the following sub-sections, following the order of execution.



(a) Primitives. (b) PO-graph. (c) Partitions.

(d) Per-partition trees. (e) Merged tree.

Figure 2: The search space partitioning pipeline.

5.1 Primitive Overlap Graph Generation

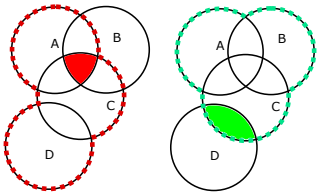
For expressing spatial relationships between primitives, the Primitive Overlap (PO)-graph is introduced. It represents spatial overlap between primitives using an undirected graph $G = (P, O)$, where $P = \{p_1, \ldots, p_{n_p}\}$ is the set of n_p primitives as vertices and O is the edge-set that contains 2-tuples of overlapping primitives $o = (p_i, p_j)$, where $i, j \in \{1, \ldots, n_p\}$ with $i \neq j$.
The PO-graph is generated based on the location, orientation and geometric shape of the primitives, see Fig. 2b. Complex shapes can be approximated with

ISBN 978-80-86943-41-1

simpler bounding volumes like Oriented Bounding Boxes (OBBs) or the convex hull of the corresponding point-set [PH77].
For better scalability, the computational complexity can be reduced from $\mathcal{O}({n_p}^2)$ (overlap check between each primitive and each other primitive) to $\mathcal{O}(n_p \log(n_p))$ using hierarchical space partitioning schemes like e.g. Octrees [Mea82].

5.2 Search Space Partitioning

With known primitives and their spatial relations given by the PO-graph, the goal is now to find independent search space partitions.
A partition is a set of primitives in which each primitive has an overlap with each other primitive. In this context, independence means that per-partition solutions are not influenced by the solutions of other partitions. See Fig. 3 for explanatory examples.
The problem of finding all independent search space partitions is equivalent to the problem of finding all maximum complete subgraphs (maximum cliques) in G. For finding the set of maximum cliques in G, the Bron-Kerbosch Algorithm (BKA)[BK73] is employed due to its behavior on random graphs: It was experimentally shown in [BK73] that the computational complexity of BKA is almost independent of graph size for random graphs. In a worst case scenario (using Moon-Moser Graphs [MM65]), computational complexity is proportional to $(3.14)^{\frac{n}{3}}$, where n is the size of the graph. Note that, if there is only a single partition for a particular PO-graph, the search space partitioning method degenerates to standard GA-based CSG-tree extraction.



(a) Incorrect. (b) Correct.

Figure 3: In (a), per-patition solution parts containing A and C are partially influenced by B (red area) but B is not part of the partition. In (b), D is not part of the partition and influences C only in an area (green) that does not overlap with other partition members. Thus, per-partition solutions are not influenced by D.

5.3 Per-Partition CSG-Tree Extraction

With known partitions, CSG-tree extraction is conducted for each partition separately in a divide-and-conquer manner. A variant of the GA described in [FP16] is used with the objective function

$$E(t) := \sum_{i=1}^{|S|} \left\{ e^{-d_i(t)^2} + e^{-\theta_i(t)^2} \right\} - \alpha \cdot \text{size}(t), \qquad (1)$$

where t is the tree candidate, S is the point-set corresponding to the partition's primitives and size(t) is the number of nodes in tree t weighted by a factor α. $d_i(t) = \beta \cdot f_t(s_i)$ is the signed distance between point s_i and the surface defined by tree t weighted by a factor β. $\theta_i(t) = \gamma \cdot \arccos(\nabla \hat{f}_t(s_i) \cdot n_i)$ is the angle between the point normal n_i and the normalized gradient at position s_i weighted by a factor γ. α, β and γ are user-controlled parameters. The first term in Equation 1 (under the sum) estimates how close the surface induced by t matches the point cloud, while the second term penalizes trees with a large number of nodes. The given objective function has to be maximized for t.
Initially, the population T_0 is filled with n_T randomly generated trees with a height $\leq h_{max}$. For the maximum tree height, the approximation

$$h_{max} \approx \sqrt{\pi/2 \cdot n_{pp} \cdot (n_{pp} - 1)} \qquad (2)$$

is used, where n_{pp} is the number of primitives in the partition. It is based on the average height of binary trees for a given number of internal nodes [FO82] and achieved good results in all experiments carried out.
Each GA iteration i contains the following steps:

1. The population of the previous iteration T_{i-1} is ranked according to Equation 1.
2. The current population is initialized with the n_b best candidates from T_{i-1}.
3. As long as T_i has not reached maximum population size n_T, two candidates are selected from T_{i-1} via Tournament Selection parameterized with k_{ts} (the size of the set of randomly chosen population members from which the best member is selected) [MG95]. During crossover, the two selected candidates exchange randomly selected subtrees with a probability of γ_{cr}. Then, with a probability of γ_{mu}, each resulting tree is mutated. Either a randomly chosen subtree is replaced with a new randomly generated subtree with a probability of μ_{mu}. Or, with a probability of $1 - \mu_{mu}$, the whole tree is replaced with a new randomly generated tree.
4. The termination condition is met if the score of the best CSG-tree candidate of an iteration does not improve over n_{tc} iterations.

The most computationally expensive step in GA-based CSG-tree recovery is the evaluation of Equation 1 for each element of a candidate-set. Since evaluations can be conducted for each candidate independently, parallel processing schemes can be applied efficiently. In addition, the solution space partitioning allows for a per-partition parallelization strategy. Both options were implemented for multi-core processors. Their evaluation is discussed in Section 6.

 ISBN 978-80-86943-41-1

5.4 Merge of Per-Partition Trees

Merging all trees corresponding to partitions into a single tree is not trivial. A simple union of all tree root nodes may lead to incorrect results if primitives that are part of multiple cliques are not splitted. Split operations on arbitrary primitive shapes tend to be complex and should be avoided. See Fig. 4 for examples. The proposed merge strategy does not need splits but instead tries to merge trees with a subtree in common. Result correctness is given since no additional operations are introduced and operation order is preserved. The strategy consists of the following steps:

1. All trees are inserted in a list L without any specific order. Extracted trees might contain artefacts affecting their mergeability (e.g., intersections with the same primitive for both operands). For each tree in L, artefacts are removed by traversing the tree and replacing found patterns iteratively with their simplifications (e.g., replacing $p \cap p$ with p). The process ends if no more artefacts can be removed.

2. Two trees t_0 and t_1 are removed from the head of L, and their largest common subtree t_{lcs} is computed (with a computational complexity of $\mathcal{O}(\max(\text{size}(t_0), \text{size}(t_1)))$). The subtree's leaf-set must be a subset of the leaf-sets of both, t_0 and t_1. The largest common subtree found might exist more than once in both trees. Thus, the root nodes of each appearance of the subtree in t_0 and t_1 are stored in the lists N_0 and N_1 (see Fig. 5a).
   If t_{lcs} is empty, t_1 is appended to L and a new tree candidate t_1 is removed from the head of L. In this case, the largest common subtree search is repeated with the new t_1.

3. For each node in N_0 and N_1, we check if it is a valid merge candidate by traversing the corresponding tree (t_0 or t_1) from root node to leaves following Algorithm 1. If the node can be reached this way, it is considered a valid merge candidate. The node is then replaced by the root of the other tree resulting in a merged tree t_m. If more than one valid candidate exists, the candidate corresponding to the larger tree is replaced by the root of the smaller tree. If both trees are of the same size, the candidate of t_0 is chosen (see Fig. 5b).
   If there is no valid merge candidate, the procedure is repeated with the next smaller common subtree in t_0 and t_1. If no other common subtree exists, t_1 is replaced by a new tree candidate from the head of L. Then, the largest common subtree search and its subsequent steps are repeated with the new t_1.

4. The merged tree t_m is prepended to L.

5. The merge process continues until there is only a single node left in L. Since the model to reconstruct is connected, a pair of mergeable trees exists in each iteration. Thus, the merge process always terminates.

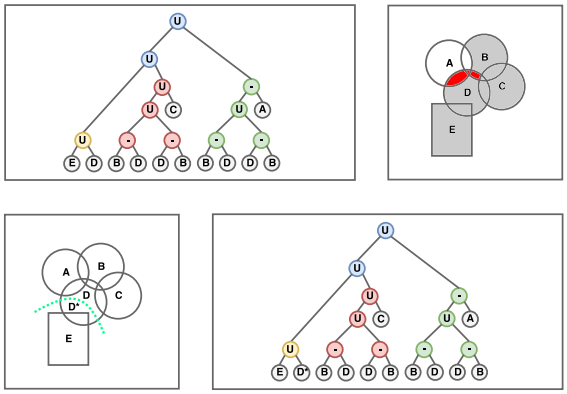
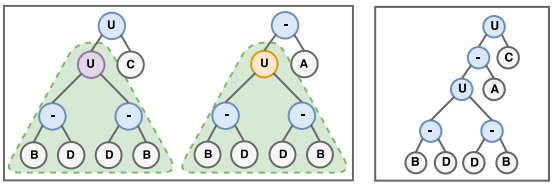


Figure 4: Merge strategies. Top: Wrong tree merge using union over all partition trees. Erroneous geometry in red (compare with Fig. 2a). Bottom: Correct tree merge using union over all partition trees with primitive splitting (green curve).



(a) (b)

Figure 5: (a) Two merge trees (t_0 left, t_1 right) with a largest common subtree (green). N_0 contains the purple node, N_1 the orange node. (b) The merged tree t_m.

```
def isValid(curNode, node):
    if curNode = node :
        return true
    if curNode.nodeType = Operation :
        if curNode.operationType = Difference :
            return
                isValid(curNode.children[0],
                node)
        elif curNode.operationType = Union :
            for child ∈ curNode.children :
                if isValid(child, node) :
                    return true
    return false
1 isValid(t.root, node)
```

Algorithm 1: Checks if node *node* is a valid merge candidate in tree t.

The merge process has an asymptotic computational complexity of $\mathcal{O}(|L|^2)$ since in the worst case L has to be traversed for each merge.

ISBN 978-80-86943-41-1

6 EVALUATION

The proposed partitioning scheme has been evaluated on a laptop with quad core CPU and 16GB of RAM on four different models. For models M0, M1 and M2, point clouds were generated by sampling a predefined CSG-model that served as ground-truth. Gaussian noise ($\mu = 0.0, \sigma = 0.01$) was added to the points to simulate measurement errors. Model M3 is based on real measurements, and primitive fitting was done with RANSAC [SWK07]. See Fig. 10 for the intermediate steps results for model M1, and Fig. 11 for point clouds and renderings for models M0, M2 and M3. Table 1 depicts model details.

	M0	**M1**
# Primitives	17	4
# Points (low)	11.3k	9.3k
# Points (high)	156.4k	158.4k
# Partitions	(0,8,4,0,1,1)	(0,0,2)
	M2	**M3**
# Primitives	29	18
# Points (low)	10.9k	-
# Points (high)	155.4k	55.8k
# Partitions	(0,0,0,12)	(0,7,4,1)

Table 1: Details on evaluated models. 'low' and 'high' indicate different sampling rates. Numbers of partitions are depicted per partition size. First position in parantheses indicate number of partitions of size 1 and so on.

The baseline is the GA approach proposed in [FP16] and described in Section 5.3. The parameter-set used for both, baseline and partitioning scheme, is listed in Table 2. The following combinations were evaluated:

- **Baseline:** Single-threaded (BST), multi-threaded GA (BMTGA).
- **Search Space Partitioning:** Single-threaded (SST), per-partition multi-threaded (SMTP) multi-threaded GA (SMTGA), per-partition and GA multi-threaded (SMTPGA) combined.

6.1 Computation Times

Timings for baseline and search space partitioning variants were measured for all models with high- and low-detail sampling (except for model M3 for which only a single point cloud exists). Measurements vary significantly for the same benchmark setting due to the stochastic behavior of GA-based methods. In order to deal with this variance, each experiment was repeated 5 times.

In the following, timing results for all methods in combination with high-detail sampling are discussed. See Fig. 6 and 7 for an overview of the results. For model M0, SMTGA is the fastest method. It outperforms the baseline by a factor of 15.3 (single-threaded, BST) and 7.5 (multi-threaded, BMTGA) on average. For model M1, search space partitioning performs worse than baseline: The fastest baseline method (BMTGA) is on average 1.4 times faster than the best-performing search space partitioning variant (SMTGA). This can be explained by the relatively small number of primitives (4) and partitions (2) in model M1, which reduces the need for partitioning. For model M2, single-threaded partitioning is 38.3 times faster than single-threaded baseline and multi-threaded partitioning variants are between 43.4 and 46.6 times faster than multi-threaded baseline. The considerable difference is due to the relatively high number of partitions (12) and their equally distributed size (all contain 4 primitives). For model M3, SMTGA is again the fastest method. Compared to multi-threaded baseline it is 3.0 times faster on average.

Search space partitioning with GA parallelization (SMTGA) is in general faster than their per-partition counterparts (SMTP, SMTPGA) for all models. This is due to the granularity and regularity of the parallelization: For SMTGA, the task of ranking a population can be splitted into n_T parts, with each part having similar execution times. For per-partition variants, granularity is determined by the (potentially lower) number of partitions, and per-partition execution times may vary significantly depending on partition sizes.

Results for per-partition variants do not show timings for the different pipeline steps since in all experiments, per-partition CSG-tree extraction is by far the most dominant factor. Timings for PO-graph generation, search space partitioning and tree merge make less than 1‰ of the total runtime.

Parameter Name	**Value**
Population size n_T	150
# Best parents n_b	2
Crossover probability γ_{cr}	0.3
Mutation probability γ_{mu}	0.3
Subtree replacement probability μ_{mu}	0.5
Tournament selection parameter k_{ts}	2
Tree size weight α	log(#pts.)
Distance weight β	100.0
Angle weight γ	$18.0/\pi$
# Iterations w/o quality increase n_{tc}	10
Maximum tree height h_{max}	see Eq. 2

Table 2: Parameters for the baseline and search space partitioning approach.

6.2 Tree Sizes and Depths

Fig. 9 contains average depths and sizes of resulting trees for baseline and partitioning variants. For the latter, tree depths have increased by 25.0% (model M1) to 285.0% (model M2) compared to the input tree, while

 ISBN 978-80-86943-41-1

for baseline approaches, an increase of 0.0% (model M1) to 125.0% (model M2) is visible. Tree sizes show similar behavior: Partitioning variants produce 46.1% (model M2) to 68.2% (model M0) larger trees, while baseline approaches increase tree size by only 0.0% (model M0) to 16.7% (model M2). Comparing tree sizes between partitioning and baseline approaches directly reveals that the former results in 25.2% (model M2) to 88.6% (model M3) larger trees.
This adverse behavior shown by partitioning variants is due to the final merge step: In each iteration, the two trees that are close to each other in the tree list and have a common subtree of at least size 1 are merged instead of the two trees with the largest common subtree of all tree pairs in the merge list. Since the focus of this work is on performance, this is acceptable. In addition, the tree optimization strategy described in Section 5.4 (step 1) was also applied to baseline results for better comparability, which has positive impact on resulting tree depths and sizes.

6.3 Scalability with Respect to Point Cloud Size

Fig. 8 depicts measurement results for the ratio

$$\frac{\#\text{points}_{high}}{\#\text{points}_{low}} : \frac{\text{duration}_{high}}{\text{duration}_{low}}, \tag{3}$$

which quantifies the dependency between point cloud size and corresponding computation times. It indicates that, for larger models (model M0 and M2), the fastest partitioning approach scales up to 1.9 times better than the best performing baseline approach with respect to point cloud size.

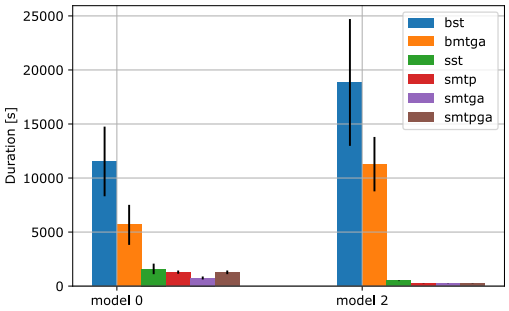


Figure 6: Timings for all approach combinations and models M0 and M2 with high-detail sampling (black lines: standard deviations).

7 CONCLUSION

In this work, a technique for accelerating an Evolutionary Algorithm for extracting a CSG-tree from a point cloud was proposed. It is based on a partitioning of the search space obtained from computing the maximum cliques of a graph of overlapping primitives, and

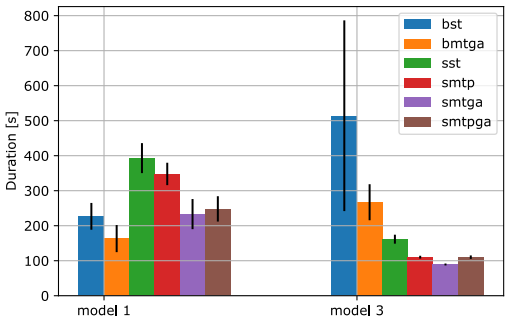


Figure 7: Timings for all approach combinations and models M1 and M3 with high-detail sampling (black lines: standard deviations).

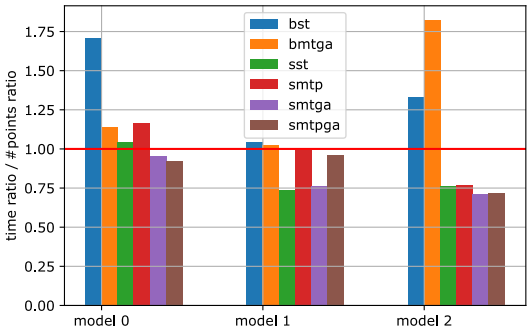


Figure 8: Ratio between high-detail and low-detail point cloud size factor and corresponding timing factors for all models (see Equation 3). The red line indicates linear scaling with a slope of 1 with respect to point cloud size. Model M3 exists only in high-detail.

on merging CSG-trees extracted for each partition. The experimental evaluation indicated a significant speedup over the baseline approach (the Evolutionary Algorithm) for different modes of parallelization.
One possible direction for future work is the implementation of the GA for massively parallel computing hardware, combined with the proposed partitioning approach. A decreased tree size in the partitioning ap-

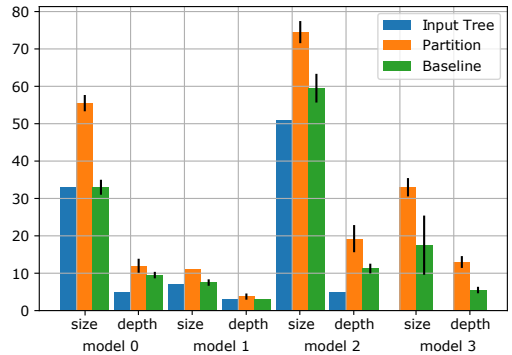


Figure 9: Average tree size and depth for baseline and partitioning methods (black lines: standard deviations).

ISBN 978-80-86943-41-1

proach could also be achieved by improving the merge process. Finally, since the partitioning (and merge) approach described in this work is independent of the technique used for the CSG-tree construction, the same approach could potentially be used with the CSG-tree conversion approaches in [SV91a, BC04].

8 REFERENCES

[BC04] Suzanne F Buchele and Richard H Crawford. Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. *Computer-Aided Design*, 36(11):1063–1073, 2004.

[BK73] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[BMV01] Pál Benkő, Ralph R Martin, and Tamás Várady. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 33(11):839–851, 2001.

[BTS+17] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, 36(1):301–329, 2017.

[ES+03] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

[FO82] Philippe Flajolet and Andrew M. Odlyzko. The average height of binary trees and other simple trees. *J. Comput. Syst. Sci.*, 25:171–213, 1982.

[FP16] Pierre-Alain Fayolle and Alexander Pasko. An evolutionary approach to the extraction of object construction trees from 3d point clouds. *Computer-Aided Design*, 74:1–17, 2016.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[KH13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, July 2013.

[Mea82] Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129 – 147, 1982.

[MG95] Brad L Miller and David E Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995.

[Mit98] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[MM65] John W Moon and Leo Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, Mar 1965.

[OBA+03] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.

[PH77] Franco P. Preparata and Se June Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.

[RC11] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[Ric73] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157–160, 1973.

[Sha01] Vadim Shapiro. A convex deficiency tree algorithm for curved polygons. *International Journal of Computational Geometry & Applications*, 11(02):215–238, 2001.

[SV91a] Vadim Shapiro and Donald L Vossler. Construction and optimization of csg representations. *Computer-Aided Design*, 23(1):4–20, 1991.

[SV91b] Vadim Shapiro and Donald L Vossler. Efficient csg representations of two-dimensional solids. *Journal of Mechanical Design*, 113(3):292–305, 1991.

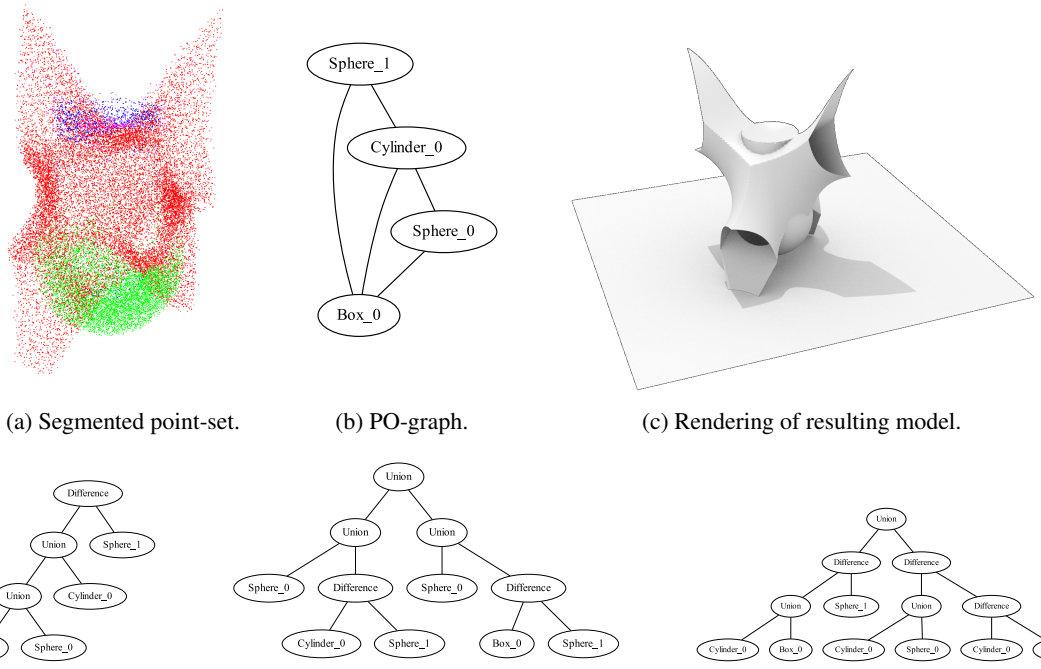
[SV93] Vadim Shapiro and Donald L Vossler. Separation for boundary to csg conversion. *ACM Transactions on Graphics (TOG)*, 12(1):35–55, 1993.

[SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer graphics forum*, 26(2):214–226, 2007.

[VMC97] Tamás Várady, Ralph R Martin, and Jordan Cox. Reverse engineering of geometric models-an introduction. *Computer-Aided Design*, 29(4):255–268, 1997.

[XF14] Jianxiong Xiao and Yasutaka Furukawa. Reconstructing the world's museums. *International Journal of Computer Vision*, 110(3):243–258, Dec 2014.

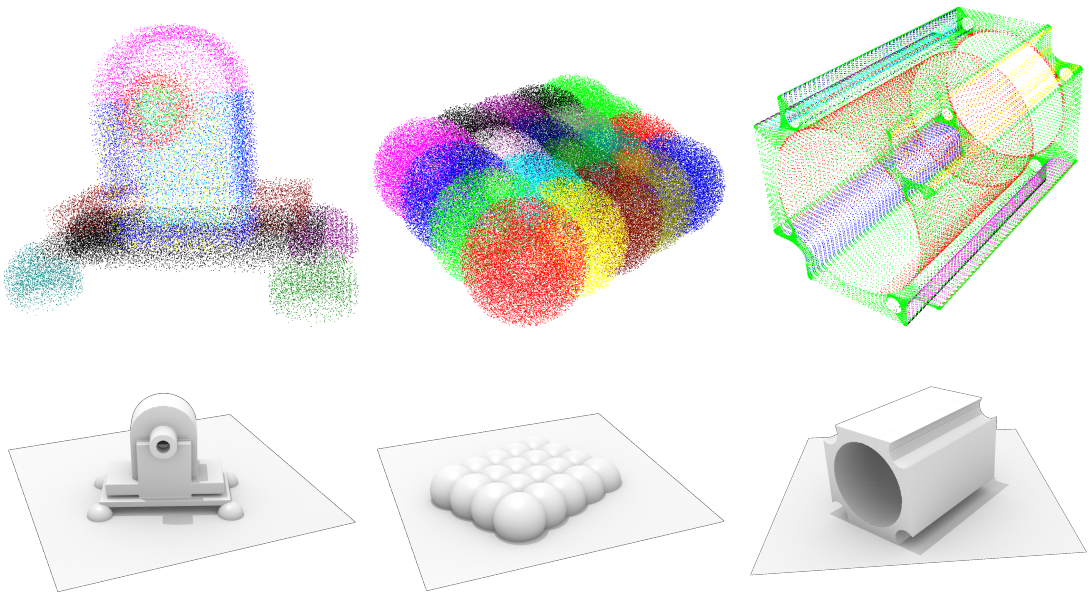
 ISBN 978-80-86943-41-1



(a) Segmented point-set. (b) PO-graph. (c) Rendering of resulting model.

(d) CSG-tree ground-truth. (e) CSG-tree from baseline. (f) CSG-tree from partitioning scheme.

Figure 10: Results of all pipeline steps for model M1. The wing-like structure is based on a simple cube whose signed distance function is distorted by a sinusoidal term. This demonstrates the flexibility of the proposed approach in terms of possible model representations.



(a) Model M0. (b) Model M2. (c) Model M3.

Figure 11: Point clouds and renderings of resulting models M0, M2 and M3.

ISBN 978-80-86943-41-1

3D underground reconstruction for real-time and collaborative virtual reality environment

Juan Manuel Jurado Rodríguez
University of Jaén, Spain
jjurado@ujaen.es

Lidia Ortega Alvarado
University of Jaén, Spain
lidia@ujaen.es

Francisco R. Feito Higueruela
University of Jaén, Spain
ffeito@ujaen.es

Abstract

Urban planning has become a relevant issue to achieve the sustainable development of the modern cities. It requires ubiquitous systems including 3D GIS capabilities and high performance without many hardware requirements. This paper deals with the development of a web application to visualize, interact and explore, through 3D perspective, the underground infrastructures of our college campus. Its 3D reconstruction raises many challenges which remains to be surmounted. The main aim of this research is the 3D modeling of underground infrastructure in order to real-time analytics of its current features. In addition, this system supports virtual reality technology to interact with the GIS data into a collaborative environment available to be performed with mobile platforms.

Keywords

Virtual Reality, Underground features, 3D GIS, Real-time visualization and interaction

1 INTRODUCTION

Urban underground space has been playing a very important role for urban development in recent years. The subsoil contains many types of features which offer a wide number of services in the city. They are usually placed at different depths so it is likely to find overlapping infrastructure elements. This is the main reason why underground must be explored through a 3D perspective in order to acquire an accurate inspection from different viewpoints of the scene. In addition, the underground infrastructures are not directly visible so their maintenance tasks become imprecise and inefficient. This trouble means challenges for 3D underground reconstruction and the extraction of its features because of its digitalization is usually outdated and incomplete.

Nowadays, three-dimensional Geographic Information Systems (3D GIS) are typically used for 3D visualization and management of virtual cities. They are applied in many professional domains like urban planning, environmental impact and resources efficiency. However, GIS users are not usually satisfied with the performance of these platforms because of the high hardware requirements for 3D rendering. In order to provide an efficient solution in this context, we have developed a custom application for 3D underground representation which includes some 3D GIS capabilities (Figure 1). Thus, this system provides complex analysis and 3D description of spatial objects and underground features. It is a web-based system which allows network access from different remote areas. Users can synchronize the current underground data and update the information system through a wireless network or a 3G access. For this reason, we have defined a client-server architecture in which the spatial database is allocated in the server and the rendering of scene is launched in the client device. The development environment in 3D is based on WebGL and the code has been optimized to be adapted to multiple mobile devices. Moreover, the usage of the WebVR API provides us support for exposing virtual reality devices in web applications. This feature has been added for underground exploring through HTC VIVE VR headset [1].

The type of source data and the fact that these features are not visible involve the challenging task of the 3D reconstruction. Currently, these infrastructures are normally depicted in 2D by means of CAD layers without any topological relationships. 2D mapping implies a high number of overlapping layers which require a upper abstraction level to identify each object. This trouble has been faced in this research work and we have applied preprocessing techniques to achieve a 3D representation. The main features of our solutions are: (1) 3D underground reconstruction, (2) virtual reality environment, (3) real-time navigation and interaction, (4) ubiquitous system adapted to different mobile devices, (5) spatial database with topological relationships. In addition, during the development of this application we have made a study with the finality of finding the best technique for modeling the terrain surface. The placement of these infrastructure is intrinsically linked to the terrain relief. Our approach is based on a precise digital elevation model (DEM) from LiDAR data used to

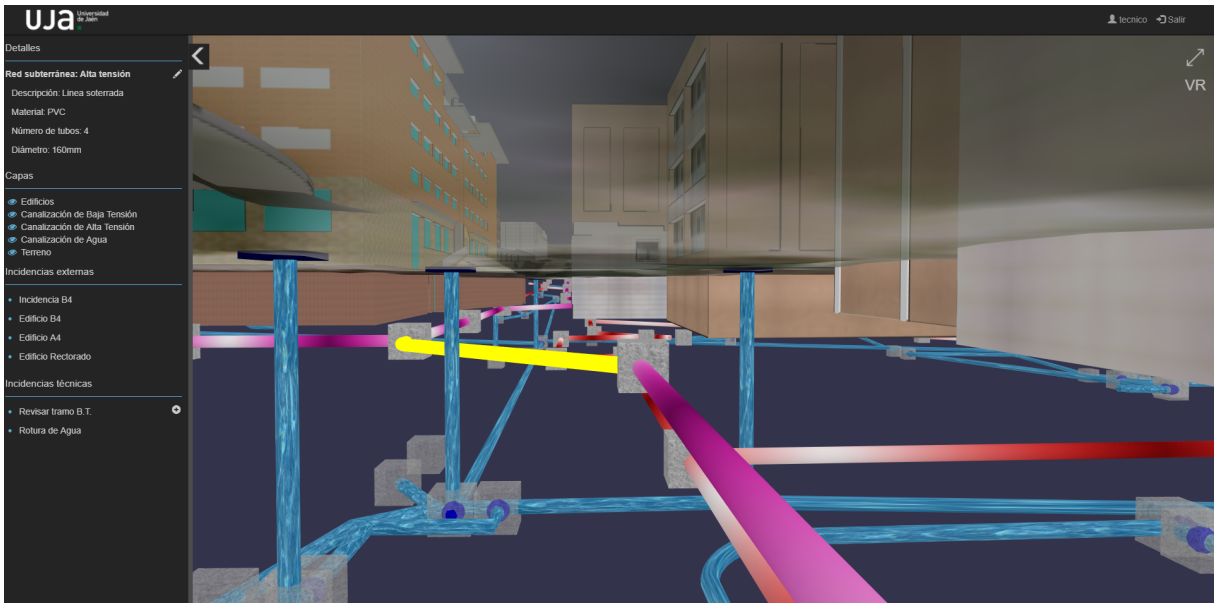


Figure 1: The 3D GIS user interface. Underground infrastructure is represented in the virtual environment. Haptic devices are used to interact with a tubing (yellow) to review his specs. Panel control is designed (left) to review the information, layers occlusion and maintenance reports.

calculate the location of the underground features according to CAD information [2].

In this paper, we first summarize the state of the art regarding GIS tools, web-based solutions for 3D mapping and acquisition methods of urban surfaces. Then, we focus on the study of the underground infrastructure before describing the steps followed for 3D reconstruction in our web-based framework. Afterwards, the results obtained and the assessment of novel contributions are discussed.

The main contributions of this paper are:

- An efficient method for 3D reconstruction of underground features and data model design.
- A web-based solution for real-time underground exploring through virtual reality into mobile platform.

2 RELATED WORK

According to 3D GIS tools there are many research projects which describe their utilities for remote sensing, forest health studies or hydraulic simulations. Recently, open source GIS [3] are known to provide user-friendly interfaces to manage and visualize 3D city models. Most of these frameworks offer a reliable representation when adding depth values to this geo-data as well as enhance their visual features. 3D GIS capabilities are focused to process large 3D models or point clouds. For this purpose, there are several frameworks and stand-alone software packages that provide some semantic and volumetric 3D models such as GRASS [4] or a popular data model like CityGML [5]. Grass includes the NVIZ visualization suite which is capable of rendering 2D/3D raster and vector maps. In this context, CityGML is a popular open standardized data model used to represent 3D cities and landscapes. The aim of CityGML is to reach a common definition and understanding of the basic entities, attributes and relations within a city model. Although both frameworks have added 3D functionality to manage city models, there are not web-based tools to efficient rendering of underground features into a collaborative workflow. In regards to proprietary GIS platforms and 3D modeling software we highlight ArcGIS [6], City Engine [7]. However, currently they are still missing some features for accurate 3D modeling, visualization and management of underground networks. However, all of these solutions are so heavy systems in order to achieve a high performance in mobile devices and the underground modeling is not supported.

Web GIS has played a relevant role in supporting collaborative environment for analysis and visualization of geospatial data on the Internet. It is considered a viable solution for gathering and sharing of collected data from various case studies [8]. The advances of web-based geo-information systems provide remote access from any location, 3D mapping and spatial analysis in real-time. In this context, the GPU acceleration is used to achieve an adequate performance for a large 3D models [9]. ArcGIS allows us to build full-featured 3D applications powered by web scenes that can include different information layers such as terrain, integrated mesh scene layers and 3D objects. Cesium [10] is a popular open-source Javascript library focused

ISBN 978-80-86943-41-1

on creating the leading web-based globe and map for visualizing dynamic data. This framework provides a complete Earth imagery support and the capability to visualize 3D models in virtual environments. Another interesting web framework is iTowns [11], which provides a visualization of 3D geographic data and precise measurements. This project supports different types of data allowing the visualization of street view images and terrestrial LiDAR point clouds. Nevertheless, all of them require so high hardware requirements to render the large size of 3D models and point clouds. Moreover, their low performances in mobile devices and the bad usability of user interfaces are enough reasons to reject the use of them for our solution.

The quality of 3D GIS tool depends on the proper choice of a precise acquisition method for surface reconstructions. Traditional techniques for creating Digital Elevation Model (DEM) are very costly regarding time consuming because of the land surveying. Today, LiDAR data and photogrammetry techniques has become one of the major methods to generate 3D ground models. Recently, LIDAR sensors on board the UAVs have become a powerful way to produce a DEM files due to the very effective data acquisition by the small distance sampling. In this paper, three acquisition methods have been explored to get a precise digital elevation model: terrestrial laser scanning (TLS), photogrammetry technique using aerial imagery that is captured with UAV (Unmanned Aerial Vehicle) and LiDAR-PNOA data provided by the Institute Geographic National [12]. Firstly, TLS stations has great potential in creating a high resolution and dense point clouds of the ground surface. However, in urban spaces there are a huge amount of buildings with their roofs and vegetation that may occlude important geometric features which require of manual setting and tedious capture processes. Secondly, UAVs have become common practice in getting visual information like images used in different fields application, such as 3D mapping, structure monitoring and cultural heritage documentation. These systems are equipped with optic (RGB) cameras oriented toward different angles to generate accurate elevation models. Finally, airborne Light Detection and Ranging (LiDAR) systems are also popular techniques in remote sensing area for accurate 3D reconstructions.

In this paper, we present a web application coded in Javascript/WebGL for the visualization and interaction of 3D underground infrastructure in real-time and distributed networking environment. It is based on Babylon.js engine and thus, supporting post-processing, controls, 3D models, animations and more features.

3 UNDERGROUND MODELING

The main aim of our study is to define the methodology for the 3D underground reconstruction through a web-based information system. It provides an accurate spatial features inventory and 3D tools for collaborative analysis and real-time interaction.

3.1 Topological spatial database design

The 3D inventory of underground utility has been created following a topological data model. The spatial database stores vectorial entities as well as their topological relationships. However, CAD files contain many objects which must be pre-processed because of their non-connected geometry. We have designed a PostGIS [13] database located in the server. It adds support for geographic objects allowing location through SQL querying. PostGIS is an extension of PostgreSQL and is released under the GNU General Public License, offering many spatial functions rarely found in other competing databases such as Oracle and SQL Server.

The underground infrastructures, that are studied in this research work, can be divided into two main layers: sewer/water and electrical wiring. In order to classify CAD files, our spatial database is composed of thematic tables that represent each underground layer. The structure defined in each table is formed by the follow attributes: ID (primary key), material, type of geometry (vector entities), diameter of pipes, number of tubes and the absolute location of geometry. Thus, topological relationships among underground entities allows us to identify influence areas for specific underground failures.

3.2 Processing of CAD layers

CAD files are the most common representation for designing urban infrastructures. Nowadays, there are not remote sensing technologies which offer an efficient techniques for geo-location of underground utility networks. A significant investment in the subsoil detection, positioning, and documentation management is currently in progress.

In this paper, we have studied the vectorial CAD layers and entities of the underground representation. Frequently, this information is stored in 2D CAD files. This data cannot be directly used for our 3D application due to the lack of geo-referenced maps and because of the incoherencies found in the input data. The result of processing this metadata is stored in a PostGIS database in order to support efficient spatial querying. For this purpose, firstly we have studied the input data to find an effective method to classify and process the whole CAD information.

In order to process the CAD file is necessary to make the follow steps: (1) the geolocation and classification of the underground layers, (2) the repair of each unconnected pipe (3) and finally the simplification of redundant geometry. Consequently, we manage vectorial data such as points, lines, polylines and polygons.

 ISBN 978-80-86943-41-1

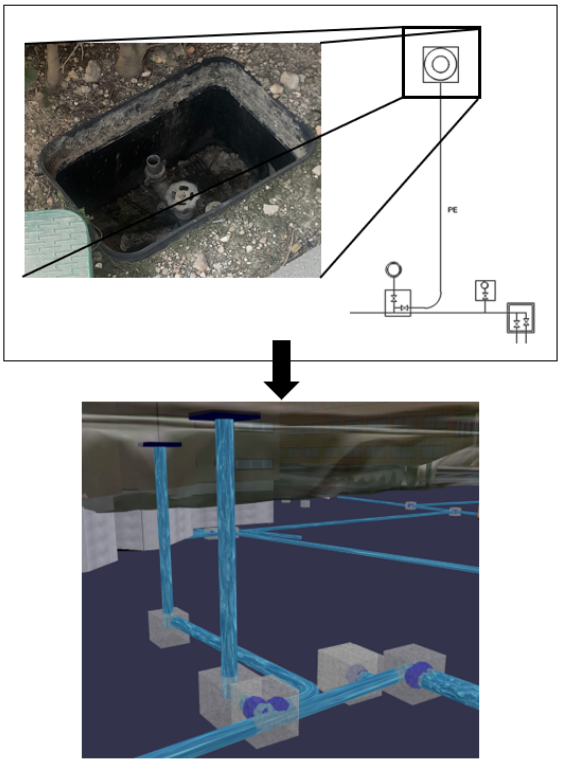


Figure 2: The enhanced visualization of our 3D GIS approach

These primitives symbolize specific entities of the real world like buildings, sewers, pipes, control stations or irrigation taps. We have used MapInfo Pro [14], which provides automatic utilities to edit the geometry and transform the vector data before being uploaded into the database server.

The input data, at first, is geo-located using three ground control points (GCPs). It makes possible the underground infrastructure mapping over the digital terrain model. Afterwards, CAD layers are classified based on the underground structure which they represent. In this work we have managed four types of entity groups: buildings, sewer network, overhead power line and low voltage wires. Secondly, the incoherent CAD objects must be repaired in order to create and manage right topological entities. Lines and polylines that represent underground pipes are usually unconnected because of drawing CAD errors. These cases are detected and solved through a method coded in MapInfo Pro. This algorithm creates new polylines if the end point of one line is close to the start point of the following. Finally, we simplify CAD layers by removing all polygons, which represent buildings, control station or some irrigation taps. They are replaced by their centroids where the corresponding 3D models are going to be subsequently located. As a result, we have made a total conversion of the original CAD map to achieve the 3D formal data structure in order to be represented in our virtual environments. Figure 2 shows the appearance improvement achieved in order to visualize the underground infrastructure reconstruction.

3.3 Digital surface modeling

Underground modeling keeps a close relationship with the relief terrain. In order to determine an accurate measure of the depth of the infrastructures, it is necessary to know the surface unevenness. In fact, our 3D GIS framework includes a Digital Elevation Model (DEM) whose vertical accuracy must be lower than 50 centimeters. A research project is carried out for generating a proper DEM of our college campus, and thus the 3D mapping of its underground infrastructure (Figure 3). For this purpose, we have used raw LiDAR data and aerial imagery from UAV.

Recently, photogrammetry has been widely used to create 3D maps and 3D models from images. We have placed a camera on the drone pointed vertically towards the ground. Multiple overlapping photos (80% overlap) of the ground are taken with UAVs through a programmed flight path. However, the terrain model is a complex and huge mesh which requires many computational efforts to be rendered in a web-based environment. In addition, there are many places of the ground which are occluded by trees and buildings. These problems have been solved using LiDAR-PNOA data. This data source is provided by PNOA project [15] which is leaded by the National Geographic Institute of Spain. This point cloud contains (X,Y,Z) coordinates, with 0.5 points/m^2, being the vertical precision lower than 20cm RMSE Z. The LiDAR information provides the capacity of calculating the Digital Elevation Model (DEM). As a result, in our application the terrain model is generated through the height map from the LiDAR point cloud. It has been calculated using Global Mapper tool in order to acquire the elevation grid surface and is exported as raster image.

3.4 Web-based 3D engine

WebGL enables a direct integration of 3D graphics into standard web pages [16]. Thereby, web applications are capable of integrating hardware-accelerated 3D graphics in network environments. It provides a co-management of heterogeneous data between client devices in order to discover, create and share 3D information. There are popular web virtual globes such as Google Maps, Apple Flyover, or OpenStreetMap [17] focused on rendering massive real-world terrain. They are composed by Digital Terrain Model (DTM), imagery and vector datasets and some 3D city landscape. The massive rendering streaming of huge 3D city models implies latency problems which must be considered in any web-based development [18]. Nevertheless,

ISBN 978-80-86943-41-1

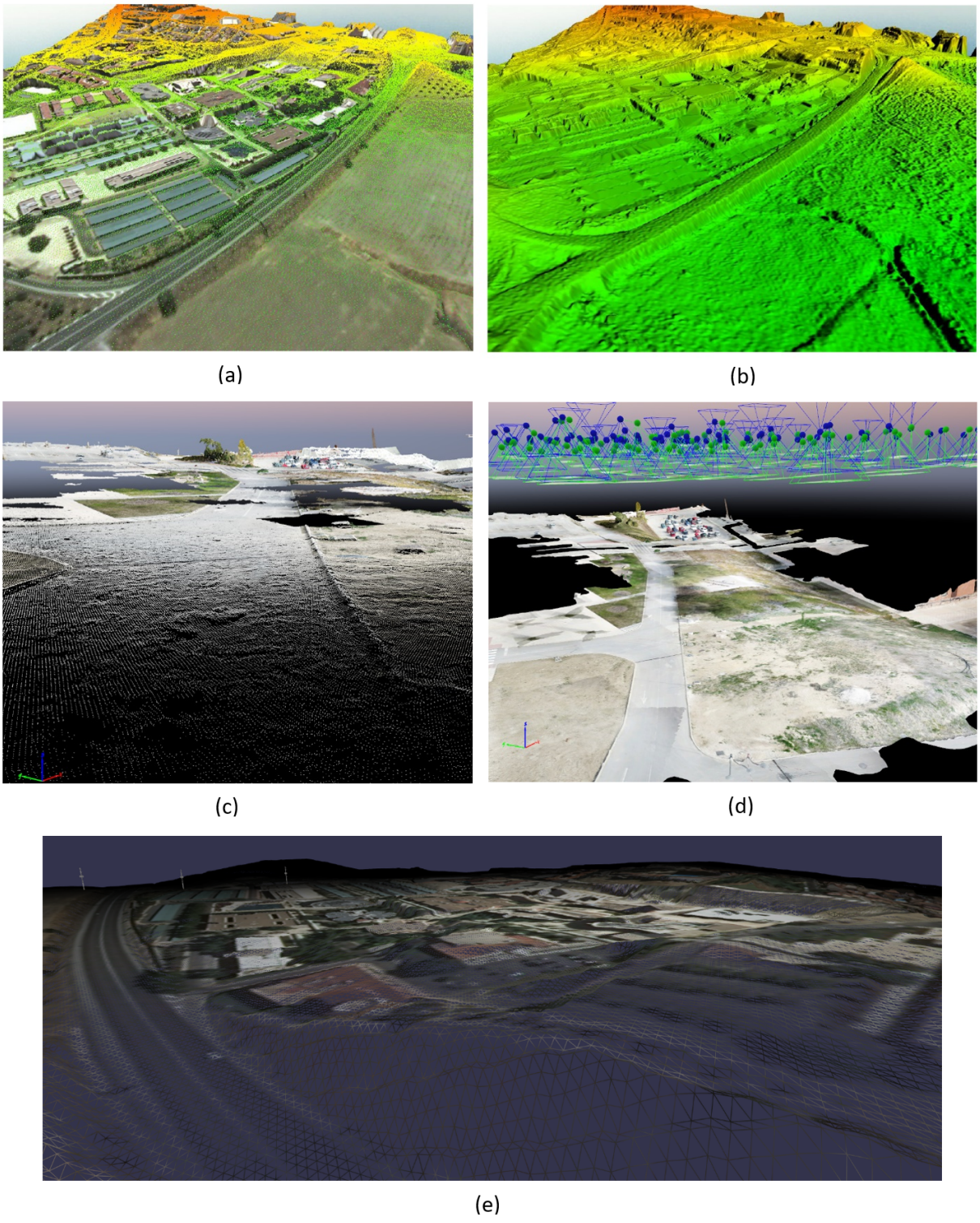


Figure 3: The relief surface modeling. LiDAR point cloud is mapped to the terrain and is processed for DEM calculation (a-b). UAV flight over the college campus and the generation of a huge mesh (c-d). As a result, the ground is calculated through LiDAR elevation model and is depicted in our application (e).

ISBN 978-80-86943-41-1

there is not any extended 3D web framework to facilitate the management of the underground structure.

The high resolution of the ground elevation model to allocate the 3D position of underground features needs efficient code for an adequate performance. The complex ground mesh rendered into our web application must be simplified in order to reduce the number of triangles generated. The 3D models in our virtual environment must be explored without lag effect, over 60 frames per second. These requirements are satisfied to guarantee the best performance into mobile platforms. Our system is based on BabylonJS engine [19]. It is an open source 3D engine coded with WebGL and Javascript to render interactive 3D computer graphics an 2D graphics within any compatible web browser. The last version of BabylonJS contains important improvements like WebGL v.2 and WebVR v.1.1 support, a better performance for invisible Solid Particle System (SPS) and PBR rendering techniques aim to simulate real life lighting. During the development of our 3D graphic scene, we have carried out the following features: the height maps to generate realistic grounds, the use of Solid Particle System (SPS), the octrees to optimize the collisions calculation and the WebVR camera to provide virtual reality navigation around the environment reconstruction.

The rendering of a huge terrain mesh requires a high computational effort to perform the scene. Instead of importing a 3D ground model, the terrain is generated by a height map. It is a grayscale image whose pixel's color is interpreted as the distance of the displacement or height from the floor. As a result, the ground is a triangulated mesh which is defined with a specific number of subdivisions. It increases the complexity of the mesh in order to improve the visual quality. In our case, depending on the target device, the ground subdivisions are changed to ensure the best performance.

Another main utility of our framework is linked to the SPS feature. It is an updatable particle system composed by separate and different geometry forms. Each particle has the same properties than any other mesh. The use of this particle system has made possible an efficient rendering of the underground infrastructures. If each pipe was represented as a unique object, the total number of meshes was very high and the frame rate dropped quite suddenly. A specific underground layer has one particle system which contains sewerage networks, electrical control stations and other 3D entities. As a result, our scene simultaneously manages until four particle systems.

The last enhance is focused on the virtual reality requirements. One of the costliest feature is the collisions detected with the 3D models of the scene. It has been optimized using a tree data structure that can improve the selection of entities based on space coordinates. In our scene buildings cannot be crossed during the navigation process, then collisions must be computed. The use of octrees optimizes the selection of sub-meshes belonging to the buildings for detecting quickly collisions. Afterwards regarding virtual reality navigation, the camera setting must be modified. Babylon 3.0 supports WebVR API 1.1 specifications in the latest version of Microsoft Edge, Chromium and Firefox. We add the functionality of VRcamera in our 3D environment to explore underground utility, through virtual reality, from any location with mobile platforms.

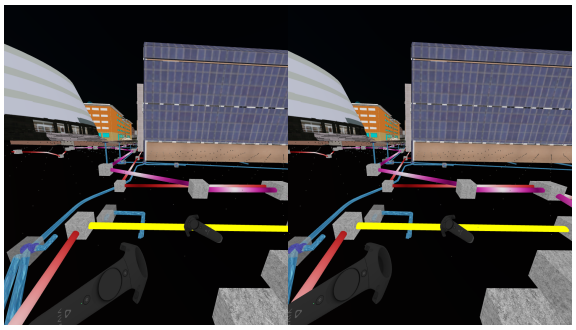


Figure 4: Virtual reality environment through HTC VIVE glasses

3.5 Bringing Virtual Reality to the Web

Real-time interaction with 3D models and 3D navigation around the scene for exploring the underground features are overcome challenges. Moreover, the efficient scene rendering allows the visualization of the 3D virtual environment without any decrease of the performance in mobile devices. Current web GIS tools provide 3D representation of spatial data, but without VR features. Virtual reality is playing an important role in many computer vision applications for a direct interaction with GIS data [20]. This technology provides immersive environments and therefore its usage for underground exploring provides a realistic interaction with the infrastructures (Figure 4). Our VR approach involves an innovative solution for visualization and analysis in the field of urban planning. It definitely helps to acquire a 3D perception of these infrastructures and plan the future urban growing, taking advantage of the complete knowledge of underground utilities.

Our framework is a web-based solution focused on real-time analytics of the 3D underground features for mobile platforms. Thus, it is necessary to apply optimization techniques to improve the performance of the rendering process, specially in virtual reality environments. In addition, some latency issues usually present in these type of web applications must be resolved. This is the reason why the meshes of the scene have been reconstructed, using simplification methods, in order to reduce the size of 3D models. In the testing process, we have used HTC Vive headset to interact and visual-

ISBN 978-80-86943-41-1

ize the 3D environment due to the perfect behavior of its haptic controllers. The VR display quality is one of the most important components of virtual reality headsets. However, the frame rate needed for rendering virtual reality applications is an important restriction. It must be high enough to prevent motion sickness and provide a smooth experience. In this context, we have applied the following optimization techniques to assure 60 frames per second (fps).

- **Virtual camera setup:** Firstly, in order to improve the performance of the 3D environment we have reduced the view field of camera. The (X, Y, Z) planes have been delimited until 30 units.
- **View Frustum culling:** In order to render only the 3D models visible by the camera we have applied frustum culling technique. View Frustum Cullers (VFCs) are typically used in virtual reality applications [21]. This method provides a significant improvement of the performance because only the 3D models inside this volume are rendered in the scene.
- **Octrees implementation:** The collision calculation requires an important computational effort [22]. In this case, the use of octrees in the 3D buildings meshes reduces the time required for the collision detection.

4 DISCUSSION OF THE RESULTS

This paper presents an innovative web-based application (Figure 5) for real-time interaction and visualization of 3D underground infrastructure of urban spaces through virtual reality. This application involves a continuous refinement model that combines an integral spatial database to store the geo-location of the subsoil objects and the descriptive information of the underground infrastructure. This system has a set of tools for 3D inspections, navigation, interaction and analysis on-site where there is any underground fault. In general, these infrastructures cannot be directly visualized. The contribution of virtual reality provides the possibility of a direct interaction with the underground features. This technology allow us to acquire a realistic perception of underground and a high knowledge of its current features. For this, we have chosen HTC Vive headset due to the accurate interaction of its haptic controllers. As a result, the immersive experiences, during the underground exploration, raises a novel way to analysis these infrastructures. In addition, we have described different acquisition methods for ground model generation. LiDAR-PNOA has been the data resource chosen for our application. The LiDAR point cloud has a high horizontal and vertical accuracy and provides a precise height map which is required to create the ground model in our application. It makes possible an accurate

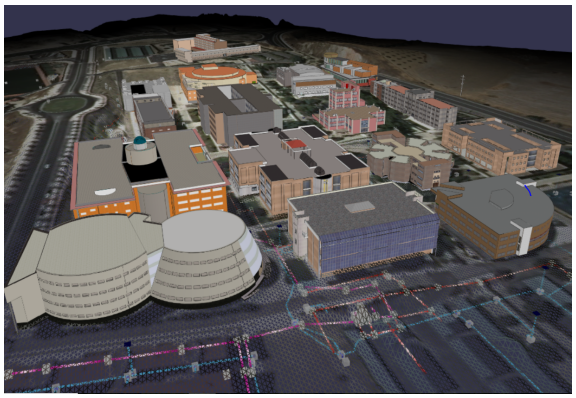


Figure 5: 3D GIS environment

3D mapping of underground infrastructure with the terrain relief and the calculation of its depth. In this paper, we have studied the input CAD information, the maintenance reports and we have generated an accurate ground elevation model our college campus. Based on this data, we have developed a 3D virtual environment to represent and manage the underground infrastructure providing as well a collaborative workflow. Thus, the users can also realize of the changes carried out by others at the same moment, due to the real-time update of the database information available in the application.

5 CONCLUSIONS AND FUTURE WORK

Underground infrastructure is the focus for many urban planning studies. Its facilities form the backbone of the progress and welfare of the modern cities. In the last few years, there has been a growing recognition about the benefits due to the accurate geolocation of underground infrastructure. The deterioration of these utilities needs a routine maintenance which must be proactive rather than reactive emergency response. The study and analysis of the growing underground facilities are challenges of many research works due to the complex data structure and the inability of their direct inspections. The resource optimization and the correct planning of resources consumption in the cities are some of the current issues that must be analyzed. In this paper, we have described the features of our web-based system in order to share the effective methodology followed for 3D underground management. The main goal is its 3D reconstruction in order to real-time interaction and study its current features.

In this paper, we propose an innovative 3D environment which provides a real-time interaction, visualization and management of underground infrastructure into a collaborative web-based application. During many years, we have studied the underground features and the best way to monitor them. Today, we show a system based on WebGL which has been optimized to

 ISBN 978-80-86943-41-1

be performed from mobile platforms. In addition, it includes a spatial database in order to store the whole descriptive information of the underground features. Our three-dimensional web-based system opens new lines for collaborative communication between planners and provides a reliable interaction with the underground features through virtual reality. Currently, we are working in a continuous increment of 3D underground representation to support 4D analysis and thus predicting the future needs for the smart city maintenance.

6 ACKNOWLEDGMENTS

This work has been partially supported by the Ministerio de Economía y Competitividad and the European Union (via ERDF funds) through the research projects TIN2014-58218-R and TIN2017-84968-R.

7 REFERENCES

[1] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.

[2] V. Meesuk, Z. Vojinovic, A. E. Mynett, and A. F. Abdullah, "Urban flood modelling combining top-view lidar data with ground-view sfm observations," *Advances in Water Resources*, vol. 75, pp. 105–117, 2015.

[3] S. Steiniger and A. J. Hunter, "Free and open source gis software for building a spatial data infrastructure," *Geospatial free and open source software in the 21st century*, pp. 247–261, 2012.

[4] M. Neteler, M. H. Bowman, M. Landa, and M. Metz, "Grass gis: A multi-purpose open source gis," *Environmental Modelling & Software*, vol. 31, pp. 124–130, 2012.

[5] T. H. Kolbe, "Representing and exchanging 3d city models with citygml," in *3D geo-information sciences*. Springer, 2009, pp. 15–31.

[6] L. F. Marques, J. A. Tenedório, M. Burns, T. Romão, F. Birra, J. Marques, A. Pires *et al.*, "Cultural heritage 3d modelling and visualisation within an augmented reality environment, based on geographic information technologies and mobile platforms," 2017.

[7] S. P. Singh, K. Jain, and V. R. Mandla, "Image based virtual 3d campus modeling by using cityengine," *American Journal of Engineering Science and Technology Research*, vol. 2, no. 1, pp. 01–10, 2014.

[8] M. D. Crossland, B. E. Wynne, and W. C. Perkins, "Spatial decision support systems: An overview of technology and a test of efficacy," *Decision support systems*, vol. 14, no. 3, pp. 219–235, 1995.

[9] M. Heitzler, J. C. Lam, J. Hackl, B. T. Adey, and L. Hurni, "Gpu-accelerated rendering methods to visually analyze large-scale disaster simulation data," *Journal of Geovisualization and Spatial Analysis*, vol. 1, no. 1-2, p. 3, 2017.

[10] B. He, W. xiong Mo, J. xing Hu, G. Yang, G. jun Lu, and Y. q. Liu, "Development of power grid web3d gis based on cesium," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Oct 2016, pp. 2465–2469.

[11] O. IGN. (2018, feb) itowns. [Online]. Available: http://www.itowns-project.org/

[12] G. Vosselman and H.-G. Maas, *Airborne and terrestrial laser scanning*. CRC Press, 2010.

[13] R. O. Obe and L. S. Hsu, *PostGIS in action*. Manning Publications Co., 2015.

[14] P. Bowes, "Mapinfo pro™-desktop gis," 2016.

[15] T. Hermosilla Gomez *et al.*, "Detección automática de edificios y clasificación de usos del suelo en entornos urbanos con imágenes de alta resolución y datos lidar," 2011.

[16] M. d. l. Calle, "Glob3 mobile: hacia un sig 3d para entornos apple-ios, android y webgl," 2012.

[17] L. Yu and P. Gong, "Google earth as a virtual globe tool for earth science applications at the global scale: progress and perspectives," *International Journal of Remote Sensing*, vol. 33, no. 12, pp. 3966–3986, 2012.

[18] Q.-D. Nguyen, M. Bredif, D. Richard, and N. Paparoditis, "Progressive streaming and massive rendering of 3d city models on web-based virtual globe," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2016, p. 83.

[19] J. Moreau-Mathis, *Babylon. js Essentials*. Packt Publishing Ltd, 2016.

[20] Z. Lv, X. Li, and W. Li, "Virtual reality geographical interactive scene semantics research for immersive geography learning," *Neurocomputing*, vol. 254, pp. 71–78, 2017.

[21] U. Assarsson and T. Moller, "Optimized view frustum culling algorithms for bounding boxes," *Journal of graphics tools*, vol. 5, no. 1, pp. 9–22, 2000.

[22] C. Tzafestas and P. Coiffet, "Real-time collision detection using spherical octrees: virtual reality application," in *Robot and Human Communication, 1996., 5th IEEE International Workshop on*. IEEE, 1996, pp. 500–506.

 ISBN 978-80-86943-41-1

Actor 3D reconstruction by a scene-based, visual hull guided, multi-stereovision framework

Muhannad Ismael[1,2], Raissel Ramirez Orozco[1], Céline Loscos[1], Stéphanie Prevost[1],Yannick Remion[1],
[1] CReSTIC-RVM lab,University of Reims Champagne-Ardenne,France,<firstname.lastname>@univ-reims.fr
[2]WISIMAGE, Clermont Ferrand, France,<firstname.lastname>@wisimage.com

Abstract

This paper proposes a novel framework to produce 3D, high-precision models of humans from multi-view capture. This method's inputs are a visual hull and several sets of multi-baseline views. For each such view set, a surface is reconstructed with a multi-baseline stereovision method, then used to carve the visual hull. Carved visual hulls from different view sets are then fused pairwise to deliver the intended 3D model. The contributions of this paper are threefold: (i) the addition of visual hull guidance to a multi-baseline stereovision method, (ii) a carving solution to a visual hull from an interpolated and smooth stereovision surface, and (iii) a fusion solution to merge differently carved volumes differing in several areas. The paper shows that the proposed approach helps recovering a high quality carved volume, a 3D representation of the human to be modelled, that is precise even for small details and in concave areas subjected to occlusion.

Keywords

3D reconstruction, shape from silhouette, multi-baseline stereovision, visual hull

1 INTRODUCTION

This paper presents a solution to 3D reconstruction with constraints set by the broadcast industry with economically sustainable 3D post-production capabilities [1]. It aims at providing a new "virtual cloning" system of actors based on multi-video capture, natively delivering full 4D textured models of actors' performance.

Modelling of 3D objects from multiple views remains a major research problem in computer vision. Several techniques such as multi-stereovision, shape-from-silhouette, shape-from-shading, and structured-light 3D scanner have been proposed for 3D reconstruction. They are usually classified as active or passive reconstruction. Active reconstruction requires controlled illumination such as a laser or a structured light, which enable high precision 3D modelling. Whereas passive reconstruction relies only on the information contained in captured images, is less restrictive on the movement of the actors, and offers the possibility of capturing actual textures. In our case, passive reconstruction is preferable as live shooting of actual performances makes controlled illumination not desirable for our 4D textured model reconstruction.

In this paper, we propose a new passive multiview approach which aims at reaching the visual quality and the precision of active approaches. Our method merges results from shape-from-silhouette and multiple multi-baseline stereovision reconstructions. Multiview or multiocular stereovision methods such as [2, 3] conveniently reconstruct surface details and concave regions. However, they fail for textureless surfaces or repetitive textures because their core computational process relies on image texture. Shape-from-silhouette methods such as [4, 5] are very useful for real time applications in multi-camera environments [6] and handle conveniently textureless and specular surfaces. However, their reconstruction quality is somehow limited as the produced visual hull (VH) cannot recover concave regions laying inside every silhouette beam. Thus, multi-stereovision and shape-from-silhouette are complementary to each other and numerous hybrid methods have already been published (see section 2).

This paper is organised as follows. Relevant previous work related to silhouette-based and stereovision reconstruction is described in section 2. Our solution builds upon a multi-baseline stereovision framework overviewed in section 3.1. The acquisition system and geometry are presented in section 3.2. The main contributions are developed in the following sections. Section 3.3 describes an adapted version of a multi-baseline stereovision framework [7] to encompass VH guidance in order to enhance its performances. Section 3.4 exposes the VH carving process, from the necessary interpolation and smoothing of raw multi-baseline results with integer disparities to the VH carving from a float-

ing point disparity map. Section 3.5 explains our process for merging the carved volumes obtained from all multiscopic units into a 3D omnidirectional model of the actor. Finally, experimental results and conclusions are discussed in sections 4 and 5.

2 RELATED WORK

3D reconstruction methods combining shape-from-silhouette with stereo can be sorted into three groups.

(i) Stereovision methods guided by visual hull Seitz and Kutulakos [8, 9] propose to build a VH and carve it according to the photo consistency of each voxel on its external surface. After a VH process, surface voxels are iteratively eliminated if they project for each view on pixels of different color. It has the benefits to model occlusions and to achieve real time reconstruction. Unfortunately, the regular space discretizing scheme leads to sampling, aliasing artifacts and partial voxel occlusions. Matsuda *et al.* [10] propose direct carving to avoid local optima. They classify the points extracted from stereovision as either credible and not credible. The VH is then carved by the credible point cloud that verifies properties. For instance, credible point normals should not significantly differ from the VH normal of the nearest point on VH surface. However, this condition is not reliable for objects with steep concavities. Li *et al.* [11] propose to use polyhedral VH to improve a stereovision-based 3D reconstruction by quality, deleting outliers. However, this method does not handle known stereovision difficulties: textureless or specular surfaces and repetitive textures.

(ii) Energy function guided, model deforming using information provided by shape-from-silhouette and stereovision This class is concerned by deformation methods (e.g. snake) exploiting concurrently the information derived from silhouette-based reconstruction and stereovision [12]. Hilton *et al.* [13] optimize the deformation of a generic mesh model of a human shape to minimize an energy function encompassing the constraints of the VH and stereovision. The main drawback of it lies in its chosen model shape and topology dependent reconstruction. It does not consider actual performance specificities such as posture (self contacts), garment (loose clothes) or physical interactions with objects or other actors. Such restrictions in shape and topology assumptions are not desirable for our project.

(iii) Collaborative methods applying simultaneously criteria borrowed from VH and stereovision techniques Song *et al.* [14] adjust a point cloud extracted from stereovision using VH information. Their method groups the VH voxels into three classes: (1) voxels containing stereovision point(s), (2) voxels intersecting a segment between such a point and the optical centre of the stereovision reference image, (3) all remaining voxels, which are assumed to represent low texture or occluded areas. A point cloud is built from first and third voxel groups and only voxels which occlude stereovision points in the reference image (group 2) are carved out. The methods in [15] and [16] are relying on Kinect sensor which has a practical limiting range of (1.2 to 3.5 m) distance.

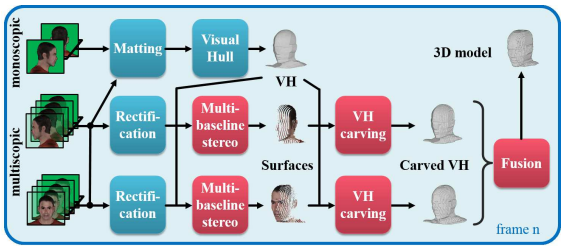


Figure 1: Proposed 3D reconstruction pipeline. Red blocks refer to specific contributions of the paper

3 PROPOSED FRAMEWORK

The proposed framework is summarized in figure 1 and borrows ideas from classes (i) to (iii). After its computation, the VH guides each multi-stereovision process per multiscopic unit. Then VH carving from stereovision is performed for each multiscopic unit similarly to class (i) but relies on a more global stereovision result close to the class (iii) concept. Finally, multiple (one per multiscopic unit) VH/multi-stereovision results are merged into a single global 3D model. Beyond its cross classification, our framework is innovative among each class. For each multiscopic unit a global scene-based multi-baseline stereovision process is run in disparity space which totally avoids partial occlusions and yields a robust stereovision result replacing more local and noisy photo-consistency usually used in class (i) carving. The proposed VH guidance class (i) is dedicated to our multi-baseline stereovision framework [7] which it enhances in terms of domain size, outliers avoidance, and, more innovatively, robustness in multi-stereovision similarity. The class "VH carving from stereovision" (iii) relies on voxel classification for voxels occluding the stereovision solution (group 2 in [14]). This classification is usually based on rays from the surface to the reference image. Replacing this image-based classification by a volumetric one in disparity space brings more precision and robustness to our solution. Furthermore, the multiple carved VH are merged at final stage. A smart handling reconstruction of inconsistencies from separate multiscopic units, conveniently corrects some residual stereovision mismatches.

3.1 Underlying multi-stereovision framework

3.1.1 Overview

This paper builds upon the VH guided multi-baseline stereovision process of Ismael *et al.* [7] for multi-

ISBN 978-80-86943-41-1

baseline stereo-vision, illustrated in figure 1. It relies on the assumption that the n views provided are synchronized images respecting the simplified multi-epipolar geometry (parallel optical axes and converging lines of sight, see [7]). The views are numbered 0 to $n-1$ from left to right facing the scene. The main features are twofold. Firstly, the solution is searched upon its natural domain in the disparity space (DS) introduced by [17], an efficient scene sampling scheme available thanks to simplified epipolar multiscopic geometry (see figure 2 that will be detailed next section). Secondly, this solution is formulated as a *materiality* map defined on this domain, expressing for each sample point its likelihood (in range $[0,1]$) of lying on a visible object surface as a perceived (indirect) light emitter. In the following, we reformulate specific parts of this method [7] important to understand the remaining of the present paper.

3.1.2 Scene space sampling scheme

Contrarily to numerous image-based approaches, this framework is deliberately scene-based as it works wholly and directly in a discrete 3D space laid in front of a multiscopic unit. This *workbench* space expresses directly the solution domain (see figure 2) on which several relevant properties are mapped. It is defined as a set of 3D points called *target points* and defined as the intersections of pixel rays from views of adjacent cameras of the multiscopic unit in a simplified geometry configuration.

Such points are aligned on constant depth planes as illustrated in figure 2 where f is the common virtual focal length of the cameras (the actual focal length divided by the horizontal pitch) and b their interocular distance. In any such plane, every point projects on any successive views on pixels of a same horizontal shift. This common column index shift of the projections is called disparity δ and is specific to the plane. A disparity δ is an integer value, defined as the common difference of column indices of the projections and is related to the depth z of the plane by $\delta = f \cdot b/z$. In figure 2, see the point circled in red whose pixels in images 2 and 3 are distant of $b-\delta$.

Any 3D target point may thus be defined by the intersection of a plane π_δ with a constant disparity δ with the ray which goes through its pixel projection $\mathbf{p}_i$ of any image i. Hence, each target point $\mathbf{T}$ may be indexed by a DS index $\mathbf{t} = (\mathbf{p}^t, \delta)^t$. $\mathbf{p} = (u,v)^t$ is the index of the pixel on which $\mathbf{T}$ projects in a chosen reference view of index i_0 (we usually choose $i_0 = 0$). According to simplified geometry, a target point indexed by $\mathbf{t} = (u,v,\delta)$ projects into any image i of the multiscopic set on $\mathbf{p}_i \in \mathbb{Z}^2$ identified by equation 1:

$$\mathbf{p}_i \equiv (u_i, v_i)^t = \mathbf{p} + (i_0 - i)\delta.\mathbf{u} = (u + (i_0 - i)\delta, v)^t \qquad (1)$$

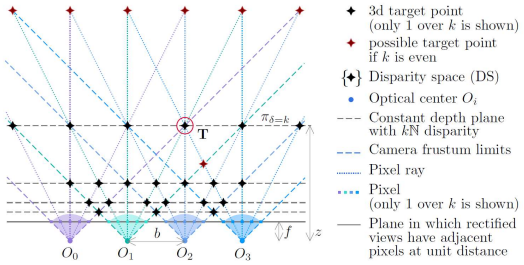


Figure 2: Disparity space: an efficient discrete reconstruction space. For clarity, only 1 over k pixels, associated rays, and constant depth planes are actually drawn.

3.1.3 Main framework concepts

Visibility: visibility reasoning evaluates for each target point with the function proposed by [2] and used in [18, 19]. This function is defined in the framework as the product of non-materiality of potentially occluding samples (see [7] for more details about the visibility function formula). DS ensures that each 3D sample point (target point) precisely lies on a genuine pixel ray in each image of the multiscopic unit for which it is inside the frustum. It thus intrinsically describes semi-occlusions ($\mathbf{p}_i$ in image i domain) and totally avoids complex treatment of partial inter-sample occlusions.

Similarity and confidence: the materiality and visibilities of target points are compared to input views, according to pre-computed similarity scores of neighbourhoods of their projections in some couples of views. This rather classical similarity computation includes (i) confidence computation typically based on variances of the neighbourhoods and (ii) a normalizing step of similarities along pixel rays which yields final similarity scores in range $[0,1]$.

Optimization and binarization: the materiality map is shaped by an optimization process, minimizing a dedicated energy penalizing deviation from intended map properties (such as completeness, smoothness, thinness) and inconsistencies between materialities, visibilities, and similarities (see [7] for more details). A binarization process delivers the final result, a binary materiality map. It is standing as a volumetric direct model of the intended solution, whereas image-based methods usually deliver disparity/depth maps that have to be processed to yield the reconstructed scene.

3.2 Shooting system and geometry

3.2.1 Studio layout and processing

Our method relies on a studio [1] composed of many synchronised and time stamped cameras with a green background (see figure 3), scattered around the observed scene in order to build the VH. Several groups laid as *multiscopic units* dedicated to multi-baseline stereovision. These units are composed of four aligned

ISBN 978-80-86943-41-1

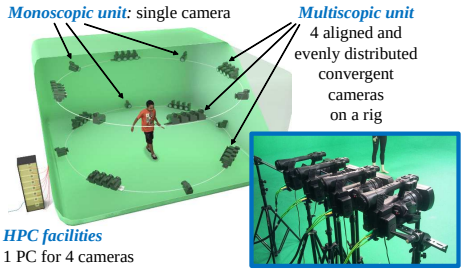


Figure 3: Dedicated multiview studio

and evenly distributed cameras. We choose a group of four which seems, according to experience, a good compromise between robustness, relying on views' redundancy, and computational efficiency (see [20]). The camera set is calibrated [21] in geometry and colorimetry in a pre-shooting step. For each time stamp, every image is matted thanks to pre-computed Chromakey and resulting silhouettes are used to compute the VH. For each multiscopic unit, captured images are then rectified to match simplified epipolar geometry.

3.2.2 VH-DS geometrical mapping

Hybridizing VH and multi-baseline stereovision implies mapping results of both methods in a same coordinate frame. Natively, VH is expressed in a regular grid in the scene frame, whereas multi-stereovision results are given in local disparity spaces, irregular in actual 3D space because their samples are not evenly spaced on fan-spread pixel rays (see figure 2).

This section presents, for any multiscopic unit, the mathematical relationship between voxel grid index $\mathbf{g} = (w,h,d)^t$ in VH, coordinates $\mathbf{r} = (x,y,z)^t$ in the frame of the rectified reference camera i_0, and coordinates $\mathbf{t} = (u,v,\delta)^t$ in DS.

This involves using (i) the VH grid parameters (scene frame reference and cell size $sw \times sh \times sd$) chosen at VH extraction step, (ii) calibration results for rectified cameras of the chosen multiscopic unit, and (iii) conversion of local depth z from reference camera i_0 to disparity δ. More precisely, we use the matrix $\mathbf{G}$ (mentioned previously in equation 2) positioning the VH grid in scene space, and the extrinsic $\mathbf{E}$ and intrinsic $\mathbf{I}$ matrices of the rectified reference camera i_0. Those matrices and their usage are exposed in equations 2, 3 and 4, where $\mathbf{R}_\Omega$ and $\mathbf{O}_\Omega$ are respectively orientation (rotation) matrices and origin points of frames Ω expressed in scene frame, and $\mathbf{Diag}(a,b,c)$ is the diagonal matrix with a,b,c values:

$$\mathbf{G} = \begin{pmatrix} \mathbf{R}_g & \mathbf{O}_g \\ & 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{Diag}(sw,sh,sd) & \\ & 1 \end{pmatrix} \quad (2)$$

$$\mathbf{E} = \begin{pmatrix} \mathbf{R}_{i_0} & \mathbf{O}_{i_0} \\ & 1 \end{pmatrix} \quad \mathbf{I} = \begin{pmatrix} \alpha_u & s & & 0 \\ & \alpha_v & & 0 \\ & & 1 & 0 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathbf{I} \times \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix} \sim \mathbf{E}^{-1} \times \mathbf{G} \times \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \quad (4)$$

The DS index $\mathbf{t}$ is thus obtained from equations 3 and 4 by adding a convenient row in $\mathbf{I}$ (in red) which adds δ, defined in section 3.1.2, to its usual $(u,v,1)^t$ output. This yields the intended equations and matrices **DSfV** and **VfDS**, transforming respectively coordinates from VH to DS (equation 5) and backwards (equation 6):

$$\begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} \sim \underbrace{\begin{pmatrix} \alpha_u & s & & \\ & \alpha_v & & \\ & & f \cdot b & \\ & & 1 & \end{pmatrix}}_{\mathbf{DSfV}} \times \mathbf{E}^{-1} \times \mathbf{G} \times \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \quad (5)$$

$$\begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \sim \mathbf{VfDS} \times \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} \quad \text{with } \mathbf{VfDS} \equiv \mathbf{DSfV}^{-1} \quad (6)$$

3.3 Stereovision guidance by VH

This section exposes how VH guidance is added and enhances performances of the multi-baseline stereovision framework [7] presented in section 3.1.

3.3.1 Core principle

In the classical VH guidance, the reconstruction solution is necessarily included in the visual hull. Indeed, any point projected outside of at least one silhouette is labelled out. It requires mapping VH and target point spaces to bounded discrete 3D grid (*cf.* eq. 5 and 6, which give real coordinates). Thus, evaluating a map $\mathbf{M}$ defined in one space for a sample of the other space is achieved via trilinear interpolation. As shown in equations 7 and 8, the interpolation is noted with angular bracketing $\langle\ \rangle$ and the mapping by round bracketing$(\ )$, whereas direct map sample evaluation uses usual square bracketing $[\ ]$:

$$\mathbf{M}(\mathbf{t}) = \mathbf{M}\langle \mathscr{U}(\mathbf{VfDS} \times (\mathbf{t}^t,1)^t)) \rangle \text{ with } \mathscr{U}((\mathbf{v}^t,a)^t) = \mathbf{v}/a \quad (7)$$

$$\mathbf{M}(\mathbf{g}) = \mathbf{M}\langle \mathscr{U}(\mathbf{DSfV} \times (\mathbf{g}^t,1)^t)) \rangle \quad (8)$$

3.3.2 Bounding DS domain

The multi-baseline stereovision framework [7] works on a 3D grid laid on disparity space DS and indexed by $\mathbf{t} = (u,v,\delta)^t$. As such, this grid has to be bounded as close as possible to useful areas where the solution is expected to stand. Without any such prior information, which is usual in purely multi-stereovision, some lateral limits are easily set in u and v according to image frustums. The disparity range is usually asked for as an input parameter delivering the missing DS boundaries. VH, defined in a bounded 3D grid, may be seen as a superset of the actual solution. Thus, the solution is in a finite and closed area of scene space generally close to the actual solution, yielding opportunities to automate and optimize the delimitation of the DS.

Projecting in DS the eight corners $\mathbf{g}_i$ of the VH grid and keeping minimal and maximal DS coordinates gives a first axis-aligned bounding box (usually abbreviated

ISBN 978-80-86943-41-1

AABB) in DS in which the solution is necessarily included. This AABB is identified by its min and max indices $\mathbf{t}_m$, $\mathbf{t}_M$ in DS as follows:

$$\left.\begin{array}{l}\mathbf{t}_m = floor(min_{i=0,\ldots,7}\ \mathbf{t}_i)\\ \mathbf{t}_M = \ ceil(max_{i=0,\ldots,7}\ \mathbf{t}_i)\end{array}\right\} \text{ with } \mathbf{t}_i = \mathscr{U}\left(\mathbf{DSfV} \times \begin{pmatrix}\mathbf{g}_i\\ 1\end{pmatrix}\right) \quad (9)$$

With no user input, this step automatizes the DS bounding. It may even optimize in lateral dimensions as the VH bounding box may appear thinner than the available views. This first AABB is further optimized according to VH information. A sweeping process is run on each of its six faces, moving them inwards as long as they contain only target points whose interpolation in VH are considered *out*. This supposes (i) that the VH is defined on the grid as a numerical map **VH** with values monotonically (let us suppose increasingly) associated to $in, surf, out$ semantics and (ii) that some interpolation threshold out_t is set. A target point indexed by $\mathbf{t}$ is thus considered out of VH according to its interpolation in **VH**:

$$\mathscr{O}ut(\mathbf{t}) = \mathbf{VH}(\mathbf{t}) \geq out_t \quad (10)$$

This double process reduces the DS domain on which the different maps are laid (allocated) which thus optimizes computational efficiency.

3.3.3 Target point filtering according to VH

Despite its computational interest, the previously described VH guidance for DS bounding eliminates only some of the potential outliers outside the final AABB. Much more outliers are to be avoided if we remember that solution samples have to lie inside VH.

A simple preprocessing step labels every target point in the optimized AABB as undoubtedly outside or possibly inside the solution according to its VH interpolation $\mathscr{O}ut(\mathbf{t})$ (equation 10). Target points labelled as outside are not given similarity scores, nor considered for matching in the multi-baseline stereovision process. They are only used as conclusively non material points for visibility reasoning purposes. This target point labelling enhances computational efficiency. It also restricts the solution domain and avoids the evaluation of some more potential outliers which directly impact the reconstruction quality as illustrated in figure 5.

3.3.4 Enhancing similarity quality

Similarity scores are computed between similar rectangular neighbourhoods in couples of views. It relies on the assumption that neighbouring pixels usually have equal disparities, and thus, that the solution is locally at constant disparity. Adaptive windowing helps to modulate this assumption according to some heuristics which may be evaluated from known data (usually pixel values) statistically expressing the assumption quality for each neighbour. We use symmetrical bilateral filtering encompassing a neighbour weight factor computed according to the colorimetric similarity to the reference pixel. For each neighbour, this weight factor is the maximum of a computation on both views. As classically stated, this enhances similarity quality.

Furthermore, the similarity computation for a target point is also enhanced by a target point labelling: as this computation implies a local constant disparity assumption, it is reasonable to exclude target points, neighbouring in the constant disparity plane, labelled outside the VH. Such neighbouring samples are filtered out of the adaptive window before similarity computation. This ensures that target points known as irrelevant do not hinder the similarity scores computation. Those similarity scores are thus more relevant, enhancing the reconstruction quality and robustness.

3.4 Carving VH from stereovision

Our visual hull voxels are labelled as *in*, *out* and $surf$. However, multi-baseline stereovision yields a surface composed of the 3D points valued 1 in the binary materiality map. Each such point also bears a final confidence score related to its confidence scores associated to its similarities and possibly, its comparison to other target points on its pixel rays. Therefore, merging both models results in the intersection between the VH and the complement of the space between the multiscopic unit and the reconstructed surface. This corresponds to the subtraction or carving from VH of the multiscopic unit to surface space.

3.4.1 Stereovision surface coding

Precisely defining the space *"between"* the reconstructed surface and the multiscopic unit is not straightforward. It is a continuous space containing and interpolating, for every view of the unit, every part of a ray going from the optical centre to any solution point which is not occluded in this view. Most of those rays are redundant across the different views. We chose, for the sake of simplicity, to replace all these view dependent segments by others, far less numerous and redundant, attached to the same solution points but coming from a single centre located at the middle of the multiscopic unit. A drawback of this simplification may lie in a loss of solution points which could become occluded in this *virtual central view*. However, as a solution point has to be seen in at least a couple of successive views, this loss does not occur when $n < 5$ because the occluding rays of a solution point are limited to 0 to $n-2$ extreme views. As such the central ray cannot be flanked by two actually occluding rays ($n = 4$) or be itself occluding the solution point ($n = 3$). This remark enforces our choice to compromise using $n = 4$.

Our surface representation is built according to a central disparity space, abbreviated as CDS, indexed in reference of the (virtual) central view. This central view is

 ISBN 978-80-86943-41-1

less biased in actual 3D space than any other and, thus, interpolation in CDS will be more relevant. According to the multiscopic geometry (see 3.1.1), it corresponds to a camera indexed $i_c \equiv (n-1)/2$. Hence, a target point of index (u,v,δ) in DS would project in the central view at $(u_{i_c}, v_{i_c}) = (u + (i_0 - i_c)\delta, v)$ (see equation 1). In order to keep integer indices for n even, we multiply the horizontal coordinate in CDS by $\gamma = 2 - n \bmod 2$. This leads to new matrices managing transformation between coordinates $\mathbf{t} = (u,v,\delta)^t$ in DS and $\mathbf{c} = (c,v,\delta)^t$ in CDS and between VH and CDS:

$$\begin{pmatrix}\mathbf{c}\\1\end{pmatrix} = \underbrace{\begin{pmatrix}\gamma & & \gamma(i_0-i_c) & \\ & 1 & & \\ & & 1 & \\ & & & 1\end{pmatrix}}_{\mathbf{CfR}} \times \begin{pmatrix}\mathbf{t}\\1\end{pmatrix} \quad (11)$$

$$\begin{pmatrix}\mathbf{c}\\1\end{pmatrix} \sim \underbrace{\mathbf{CfR}\times\mathbf{DSfV}}_{\mathbf{CDSfV}} \times \begin{pmatrix}\mathbf{g}\\1\end{pmatrix} \quad \begin{pmatrix}\mathbf{g}\\1\end{pmatrix} \sim \underbrace{\mathbf{CDSfV}^{-1}}_{\mathbf{VfCDS}} \times \begin{pmatrix}\mathbf{c}\\1\end{pmatrix} \quad (12)$$

In this central space, we decide to represent the solution surface as a disparity map **DM** tagged by a confidence map **CM**. This is achieved by assigning for each solution point in DS, from far to near, at its CDS *pixel coordinates* (c,v), its disparity δ to **DM** (initialized to $-\infty$) and its associated final confidence score to **CM**. When n is even, in order to fill gaps induced by the horizontal stretching in CDS, if two successive target points on a row of CS are both solutions, their middle point is also assigned their common disparity in **DM** and mean confidence in **CM**. No other gap may occur because the solution in CS is computed in a way to ensure that its intersection with any (u,δ) plane is a continuous sequence of adjacent target points which are of same or adjacent disparities.

3.4.2 Carving VH from disparity map

The algorithm to carve the VH according to the stereovision surface coded by **DM** and **CM** is described in 1. It aims at filling a carved volume defined as a map **CV** laid over the VH grid and valued $in, surf_0..surf_q, out$. The different $surf_i$ values refer to increasing quantified confidence levels for surface voxels. The lowest confidence level $surf_0$ is reserved for $surf$ voxels of VH that are either occluded or out of frustum for the current solution. Other levels are associated with voxels identified as $surf$ in the stereovision solution: the effective level *i* is quantified according to the interpolated **CM** value of the voxel. A key feature of this step for the latter fusion process is to yield a coherent topology to the carved volumes: *in* and *out* sets are considered in a 6-connected space while $surf_{\{0..q\}}$ is considered in a 27-connected space. With such topological evaluation, no direct 6-connection should occur between *in* and *out* voxels.

In order to handle the grid sampling while responding to the previous intended topological property, point

```
1  c ≡ (c,v,δ)   N4 = {(−1,0),(1,0),(0,−1),(0,1)}
2  foreach g in VH domain do
3      if VH[g] is in or surf then
4          c = 𝒱(CDSfV × (g^t,1)^t)
5          if (c,v) in DM domain then
6              δ_s = DM⟨(c,v)^t⟩   g_s = 𝒱(VfCDS × (c,v,δ_s,1)^t)
7              if (‖g_s − g‖_∞) ≤ 1 then
                   CV[g] = surf_{Quant(CM⟨(c,v)^t⟩)}
8              else if δs < δ then CV[g] = out
9              else
                   if VH[g] is in then CV[g] = in
10                 else CV[g]=surf_0
11                 foreach n ∈ [0,4[ do
12                     lg =
                           ‖𝒱(VfDS × (c^t + (N4[n],0,0))^t) − g‖_∞
                       n_c = (c,v)^t + N4[n]/lg
13                     if n_c in DM domain and
14                       (δ_n = DM⟨n_c⟩) < δ then
15                         cnf = (CM⟨(c,v)^t⟩(δ − δ_n)+
                               CM⟨n_c⟩(δ_s − δ)) / (δ_s − δ_n)
                           CV[g] = surf_{Quant(cnf)}
                   end
               end
16         else if VH[g] is in then CV[g] = in
17         else CV[g]=surf_0
18     else CV[g] = out
19 end
```

Algorithm 1: Carving VH by central disparity map

comparison in CDS is related to actual axis-aligned distance $\|\ \|_\infty$ in VH. Hence, the interpolated solution point $\mathbf{c}_s = (c,v,\delta_s)^t$ is projected back in VH to measure its distance to thinitial voxel $\|\mathbf{g} - \mathbf{VfCDS}\times\mathbf{c}_s\|_\infty$, with $\mathbf{g} = (w,h,d)^t$. When this distance is less than 1 (*line 7*), $\mathbf{g}$ is labelled $surf$ in the carved volume with a confidence level quantified from $\mathbf{CM} < (c,v)^t >$. If the voxel $\mathbf{g}$ is in front the surface ($\delta > \delta_s$), it is labelled *out* in **CV**. Otherwise, the voxel is *a priori* labelled *in* but could be labelled $surf_i$ if it lies close enough of a steep slope of the surface. To check this possibility, we evaluate (*line 11*) if any of its 4 neighbours in CDS of same disparity δ, at unitary distance in VH, are to be considered *out* (with interpolated disparity lower than δ). This evaluation consists in measuring the distance lg in VH of the initial voxel to a neighbour $\mathbf{n}_0$ at a unitary distance in CDS and interpolating disparity δ_n in **DM** at a neighbour $\mathbf{n}_c$ in same direction but distance lg^{-1}. If $\delta_n < \delta$ this neighbour is considered *out* and the initial voxel is re-labelled $surf_i$ where the confidence level *i* is quantified from the linear interpolation at δ of $\mathbf{CM}\langle\mathbf{n}_c\rangle$ at δ_n and $\mathbf{CM}\langle(c,v)^t\rangle$ at δ_s.

3.4.3 Improving surface smoothness

The result of multi-stereovision method leads to discontinuous surface divided into frontal planar patches with constant and integer disparity, one for each multiscopic unit (see figure 6). Removing this effect is required for the visual quality of the result (see figure 6) and for a more accurate management of reconstruction inconsistencies between different multiscopic units. To deal with

ISBN 978-80-86943-41-1

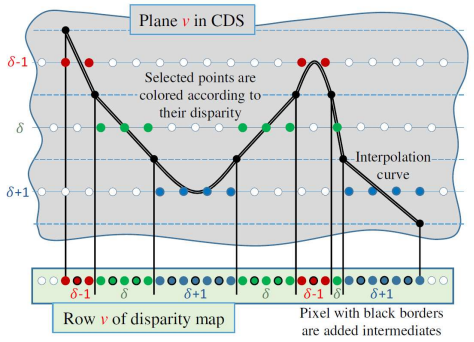


Figure 4: Disparity interpolation: relation between disparity map **DM** (coloured points) and interpolated disparity map $\mathbf{DM}_r$ illustrated in CDS by the interpolation function (black double lined curve)

this problem coming from the integer disparities quantification, we propose to represent the solution surface previously saved in **DM** by a floating point derivative version $\mathbf{DM}_r$. The map $\mathbf{DM}_r$ is computed to ensure continuous transitions between adjacent horizontal segments of constant disparities with a disparity gap of 1. Computing $\mathbf{DM}_r$ consists in looping over rows of **DM**.

Every row v of **DM** is thus scanned from one end to the other to identify disparity steps between adjacent pixels of finite disparity. When the disparity step is of magnitude $(-1,+1)$, a contact point (black point in figure 4) is placed in CDS in the middle of the two pixels with the mean of their disparity values as illustrated in figure 4, and serves as end point of both segments. Otherwise, one end point is placed for each adjacent segment in the middle of the two pixels at the segment disparity. When one of the pixels is of infinite disparity as well as for first and last pixels, a single end point is generated on the relevant pixel at its finite disparity. This process yields two end points per segment expressed in CDS $(c0,v,\delta_0)$ and $(c1,v,\delta_1)$. When a right end point $(c1,v,\delta_1)$ is generated, the corresponding segment of initial constant disparity δ is filled in $\mathbf{DM}_r$ by a dedicated interpolation scheme between the end points.

$$\mathbf{DM}_r[(c,v)^t] = \delta + (1-t)(2t-1)(\delta-\delta_0) + t\cdot(2t-1)(\delta_1-\delta) \quad , \quad t = \frac{c-c_0}{c_1-c_0} \tag{13}$$

The interpolation function in equation 13 ensures that both end points are respected (see figure 4 where the black double lined curve expresses the interpolation function producing the interpolated disparities in $\mathbf{DM}_r$). When δ_0 and δ_1 are both under or above δ, or if one equals δ, this interpolation is parabolic. When one is above and the other under, they are equal and the interpolation is linear.

3.4.4 *Smoothing using bilateral filter*

The result of the disparity interpolation described in the section 3.4.3 is a floating point disparity map more *continuous* or smooth on each row but still presenting vertically numerous depth steps. A bilateral filter is applied on the disparity map $\mathbf{DM}_r$ to compute a smoothed disparity map $\mathbf{DM}_s$ as described in equation 14 and demonstrated in figure 6. The centred operating window is chosen rectangular as regulating transitions between segments implies a rather low width $2ww+1$ but reducing vertical depth steps involves a much taller height $2wh+1$.

$$\mathbf{DM}s[\mathbf{q}] = \frac{\sum_{\mathbf{n}\in W} \mathbf{DM}r[\mathbf{p}+\mathbf{n}]\ \mathscr{W}(\mathbf{p},\mathbf{n})}{\sum_{\mathbf{n}\in W} \mathscr{W}(\mathbf{p},\mathbf{n})} \tag{14}$$

with $\mathbf{n} = (dc,dv)^t$, $W = [-ww,ww]\times[-wh,wh]$ and

$$\mathscr{W}(\mathbf{p},\mathbf{n}) = \mathscr{G}_{\sigma_c}(dc)\ \mathscr{G}_{\sigma_v}(dv)\ wd(\mathbf{DM}_r[\mathbf{p}+\mathbf{n}] - \mathbf{DM}_r[\mathbf{p}])$$

$$\mathscr{G}_\sigma(t) = \gamma_\sigma \cdot exp(-t^2/(2\sigma^2)) \qquad \gamma_\sigma = (\sigma\sqrt{2\pi})^{-1}$$

wd a function decreasing from 1, for example

$$wd(\Delta\delta) = \sigma_\delta^2/(\sigma_\delta^2+\Delta\delta^2)$$

3.5 Omnidirectional 3d modelling

3.5.1 *Merging difficulty*

The final step of the 3D reconstruction consists in merging carved VH volumes $\mathbf{CV}_m$ from multi-baseline stereovision results for all multiscopic units m in order to obtain a single 3D model representing the 3D pose of the reconstructed actor.

Figure 6 illustrates that the result of each multiscopic unit provides information only on visible surfaces facing the unit while other surface areas are left to VH result. Multiple carved VH from different multiscopic units spread around the scene thus yield stereovision details for almost every surface area of the model.

However, parts of the model surface are to be seen and reconstructed by multiple multiscopic units and those independent reconstructions are usually inconsistent one to another. Therefore, in such common areas, we have to decide which reconstruction is locally kept in the final solution. This decision is based on the confidence attribute of surface voxels: as stated in section 3.4.2, surface voxels in $\mathbf{CV}_m$ bear different labels $surf_i$ indicating their quantified confidence level according to the stereovision process.

3.5.2 *Merging process*

The overall principle of this final step is to initialize the final merged volume **FV** to one of the carved VH ($\mathbf{FV} = \mathbf{CV}_{m_0}$) and then iteratively merge each other carved VH $\mathbf{CV}_m$ into **FV** according to surface confidence decisions in differently labelled areas. As VH is known to be a superset of the solution, the process only evaluates voxels labelled *in* or *surf* in **VH**. It thus loops over every voxel **g**, treating each one for which $\mathbf{VH}[\mathbf{g}]$ is not *out* according to its labels $\mathbf{FV}[\mathbf{g}]$ and $\mathbf{CV}_m[\mathbf{g}]$:

- both *out*: voxel **g** is kept *out* in **FV**
- both *in*: voxel **g** is kept *in* in **FV**

ISBN 978-80-86943-41-1

- $surf_i$ and $surf_j$: voxel **g** is kept $surf$ with the highest confidence level $\mathbf{FV}[\mathbf{g}] = surf_{max(i,j)}$
- all other cases: voxel **g** bears inconsistent labels, the global loop is suspended while an inconsistency resolution process is run from **g**.

To decide which solution is to be kept in the last case, we propose a global evaluation of the 6-connected area implied in the detected inconsistency rather than a per voxel decision. Thus, when a voxel **g** is detected as inconsistent in the global loop, a two-pass process starts in order to make a decision.

The first pass aims at making the right decision. It goes from **g** through its inconsistent 6-connected area in order to compute the per-confidence level histograms of the encountered surfaces of both volumes. These confidence histograms for the two surfaces help making the decision on which volume **FV** or $\mathbf{CV}_m$ will transfer its labels to the final solution in this 6-connected area. We propose to choose the volume with the highest mean confidence level, but other competing scores could easily be proposed and tested from confidence histograms.

When the decision is made, a second pass is run. The same walk-through in the area is performed in order to resolve the inconsistency by copying labels of the chosen volume into the other. One could have thought that when the chosen volume is **FV** nothing needs be done, but the first pass and the decision making would then be repeated for every voxel of the area which is far from efficient. Therefore, during this second pass, when a voxel labelled $surf_i$ and $surf_j$ is encountered, its best confidence level $(max(i, j)$ is kept in both volumes.

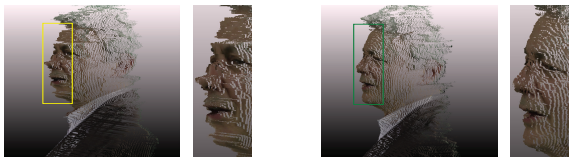
This process clearly relies on a consistent topology in both volumes. This point is ensured by the VH carving step described in section 3.4.2. This topological consistency further permits to keep our 6-connected area walk-through topologically consistent: it starts from an inside position (in or $surf_i$) in one of the volumes $\mathbf{V}_i$ and an outside position (out or $surf_i$) in the other volume $\mathbf{V}_o$. This per-volume topological position has to be ensured over the whole traversed area. No shift from *in* label to *out* label should occur in each volume across a 6-connection. Thus, ensuring topological consistency consists in avoiding 6-connections transgressing initial inside/outside position in any volume. This could occur in $\mathbf{V}_i$ for voxels on the surface connected to *out* voxels as in $\mathbf{V}_o$ for voxels on the surface connected to *in* voxels.

3.5.3 *Refinements*

A rough application of the process described in section 3.5.2 is not satisfactory because the walk-through areas sometimes appear as several,rather broad and distant, *blobs* of non surface voxels connected by thin lines or surfaces. The decision is made once for the whole area, while it should be differentiated for each blob and connection line or surface. This yields inconvenient decisions which need to be corrected. In order to do so, we apply several times the merging process of section 3.5.2 (three times in the present implementation) with less and less restrictive conditions on inconsistent voxels:

1. Considered voxels have to be labelled in/out or out/in. Furthermore a sufficient part of their 6-neighbours has to be labelled in the same way (at least 40% in our implementation). This step treats broad in/out blobs.
2. Considered voxels are the remaining in/out or out/in ones. This step treats rather thin areas.
3. Considered voxels are any other inconsistent ones. This steps finalizes the resolution and treats very thin areas with no (in, out) or (out, in) voxel.

Results from this refinement are illustrated in figure 7.



(a) (b)

Figure 5: Resulted point cloud of a real actor "Jacques". (a) point cloud obtained with integer disparity values without VH guidance and zoom in its yellow area. (b) point cloud obtained with integer disparity values with VH guided stereovision and zoom in its green area.

4 RESULTS AND DISCUSSION

To evaluate our framework described in figure 1, we used the studio layout scheme presented in section 3.2.1 both for real and virtual shooting and applied our framework to the views they produced. These experimental conditions apply to each result discussed in this section.

Figure 5 illustrates that the VH guided stereovision method described in section 3.3 improves the materiality map derived from a previous multi-baseline stereovision method [7] by ridding it of outliers outside the visual hull. Moreover, in non specular textured or concave areas, the materiality map solution proves to be more accurate than the visual hull as illustrated in first rows of figure 6 which clearly show that concavities, such as eye cavities, are carved out by our stereovision method both for virtual and actual shootings.

Figure 6 shows the results of the carving process described in section 3.4 on two view sets: the first one, of a virtual actor "Simon", shot under ideal calibration conditions by computer graphics software and the second one, of a real actor "Philippe", captured in the RECOVER 3D dedicated studio. Comparing the carved volume to the point cloud on each row of these figures, qualitatively validates our carving method. The evolutions obtained on both figures from each row to the

ISBN 978-80-86943-41-1

next, demonstrate the relevance of the disparity interpolation and smoothing steps.

The fusion of every multiscopic unit outcomes (see section 3.5) provides robust reconstruction, especially in the areas where two or more multiscopic units compete. Figure 7 demonstrates this with results obtained from a virtual and a real data set. One should notice the results' quality despite the low number of implied multiscopic units: three for the actual shooting and four for the virtual one.

To compare our results to state of the art, we apply our data (masks, RGB images, and camera parameters) to the PMVS method proposed by Yasu Furukawa [1]. We also apply chosen steps to all the results derived from multiscopic units in order to get one robust object modelling using CGAL library [2]. It includes the following steps: outlier removal, simplification to reduce the number of input points, smoothing to reduce noise in the input data, normal estimation and orientation, and Poisson surface reconstruction method.

We compare the results on the virtual data set "Simons". The first column of figure 8 shows the reconstructed visual hulls. The reconstruction using CGAL lacks overall precision, especially in the ear areas. The reconstruction using PMVS shows better results near the ear areas, but strong surface deformations, specifically at the salient parts. Our reconstruction is visually better, with smoother surface reconstruction, and specifically good results in difficult, concave regions such as the ears.

5 CONCLUSION

This paper describes a new way of combining visual hull and multi-baseline stereovision in a fully automatic process. In section 3.3, we explained how to exploit information from the VH to guide the materiality map process in order to increase its reconstruction accuracy and robustness.It was demonstrated that the materiality map framework can integrate the VH guidance in a powerful way thanks to its scene-based structure.

Our contributions are a new algorithm for VH carving from stereovision surface coded as central disparity map, and a novel framework to merge multiple carved VH obtained from different multiscopic units. This process yields a topologically consistent volume, crucial for many applications. We demonstrated on experimental examples the algorithm results, the relevance of our disparity interpolation and smoothing methods, and the efficiency of the proposed inconsistency handling on both virtual and actual shootings.

Altogether, these contributions yield a qualitative and robust omnidirectional 3D reconstruction tool. The proposed solution proves the advantages of using both multiscopic and monoscopic cameras in a studio system as well as combining multi-baseline stereovision with visual hull approaches.

[1] http://www.di.ens.fr/pmvs/

[2] https://www.cgal.org/

ACKNOWLEDGMENTS

This work was funded by the RECOVER 3D project supported by the French National Fund (FSN) for a Digital Society, and the ANR ReVeRY national fund (ANR-17-CE23-0020). We would like to thank our partners XD Productions for providing the models captured using their camera system.

6 REFERENCES

[1] L. Lucas, P. Souchet, M. Ismael, O. Nocent, C. Loscos, L. Blache, S. Prévost, and Y. Remion. Recover3d: A hybrid multi-view system for 4d reconstruction of moving actors. In *4th international conference on 3D Body Scanning Technologies, Long Beach, United States*, pages 219–230, 11 2013.

[2] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 82–96, London, UK, UK, May 2002. Springer-Verlag.

[3] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, April 2002.

[4] C.H. Chien and J.K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *CVGIP*, 36:100–113, October 1986.

[5] E.Steinbach, B.Girod, P.Eisert, and A.Betz. 3-d reconstruction of real-world objects using extended voxels. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 1, pages 569–572, September 2000.

[6] G. K. M. Cheung, T. Kanade, J. Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, pages 714–720, 2000.

[7] M. Ismael, S. Prévost, C. Loscos, and Y. Remion. Materiality maps: A novel scene-based framework for direct multi-view stereovision reconstruction. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5467–5471, October 2014.

[8] S.M. Seitz and D.R. Charles. Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vision*, 35(2):151–173, November 1999.

[9] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, July 2000.

[10] K. Matsuda and N. Ukita. Direct shape carving: Smooth 3D points and normals for surface reconstruction. *IEICE TRANSACTIONS on Information and Systems*, 2011.

[11] Ming Li, H. Schirmacher, M. Magnor, and H.-P. Siedel. Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 9–12, December 2002.

[12] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392, December 2004.

[13] A. Hilton and J. Starck. Multiple view reconstruction of people. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 357–364, September 2004.

 ISBN 978-80-86943-41-1

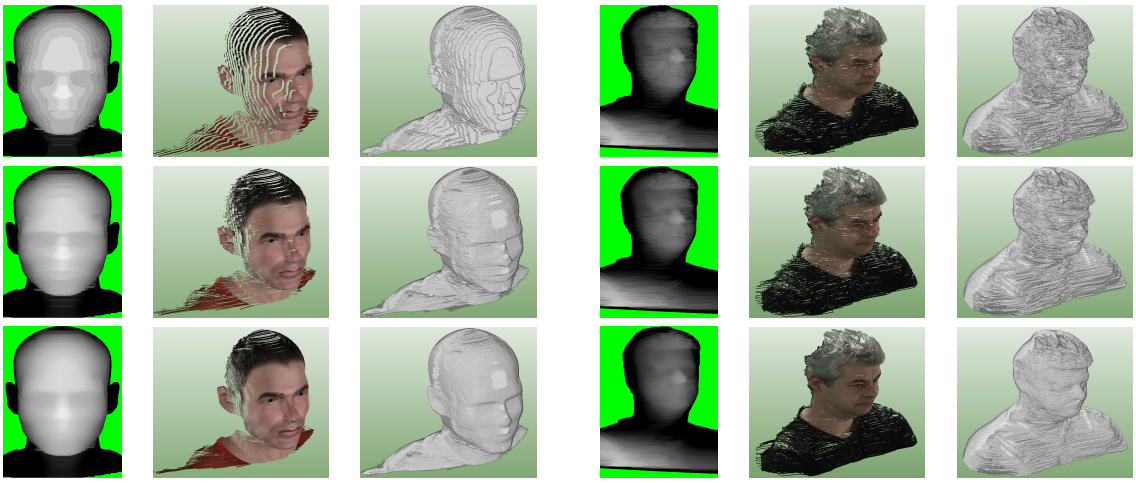


Figure 6: Results from one multiscopic unit (Left) for virtual Simon dataset, (Right) for real actor "Philippe". From top to bottom, results with: initial integer valued disparity; interpolated disparities according to 3.4.3; disparities smoothed by bilateral filtering described in 3.4.4. On each row, from left to right: disparity map, point cloud, and carved volume

Figure 7: Results of the entire pipeline. First row: several views of the point cloud and carved volume obtained from VH and four multiscopic units for virtual actor "Simon". Second row: several views of the global point cloud obtained for real actor "Jacques" from final volume resulting from VH and three multiscopic units. It corresponds to the union of the projection, per multiscopic unit, of the initial point cloud on the final volume.

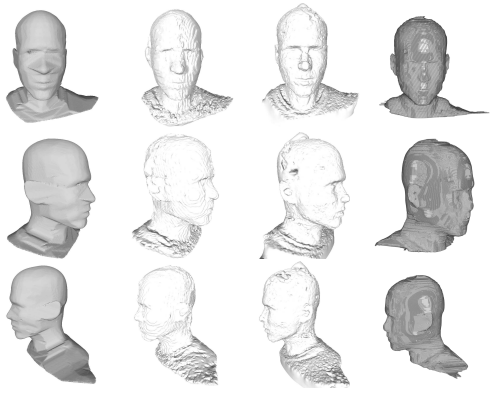


Figure 8: First column: visual hull of the ground truth virtual model "Simons". Second column: results from CGAL. Third column: results from PMVS. Fourth column: results from our framework.

[14] P. Song, X. Wu, and M. Wang. Volumetric stereo and silhouette fusion for image-based modeling. *The Visual Computer*, 26(12):1435–1450, December 2010.

[15] K.S. Narayan, J. Sha, A. Singh, and P. Abbeel. Range sensor and silhouette fusion for high-quality 3D scanning. *IEEE International Conference on Robotics and Automation, ICRA*, pages 3617–3624, 2015.

[16] R. A. Newcombe., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *Proceedings of 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[17] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '93, pages 274–279. IEEE, June 1993.

[18] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Int. J. Comput. Vision*, 32(1):45–61, August 1999.

[19] C. Niquin, S. Prévost, and Y. Remion. An occlusion approach with consistency constraint for multiscopic depth extraction. *Int. J. Digital Multimedia Broadcasting*, 2010.

[20] C. Niquin. *Reconstruction du relief et mixage réel virtuel par caméras relief multi-points de vues*. Doctorate thesis, University of Reims Champagne-Ardenne, March 2011.

[21] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence*, 14(4):407–422, Aug 2005.

ISBN 978-80-86943-41-1

Visually Realistic Graphical Simulation of Underwater Cable

Ori Ganoni
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
at time *t*
ori.ganoni@pg.canterbury.ac.nz

Ramakrishnan Mukundan
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
mukundan@canterbury.ac.nz

Richard Green
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

ABSTRACT

This paper presents different modeling considerations that are important in simulating visually realistic behavior of underwater cables attached to remotely operated vehicles. The proposed methodology has been tested on highly complex models of aquatic environments created using Unreal Engine 4. Current methods and implementations of cable simulations that are widely used in computer graphics are generally suited only to light density mediums such as air. In this paper, we present modifications to the above model required for simulating neutrally buoyant cables in underwater environments. The simulation results presented in this paper successfully demonstrate different behavioral aspects of flexible variable length underwater cables and their variations with respect to modeling parameters using our proposed method.

Keywords

Robot simulation, ROV, Underwater simulation, Cable simulation, Unreal Engine 4.

1 INTRODUCTION

Cables are used widely in underwater environments for power supply and communication to remote locations and to support various types of underwater structures. High tension cables are generally used to tow fishing equipment and research probes whereas low tension cables and ropes are used in underwater tethered systems like Remotely Operated Vehicles (ROVs) (Figure 1). The drag introduced by the water medium and the buoyancy forces make the underwater cables behave differently to less dense mediums like air or vacuum. For most ROV cables (and also in our simulation), gravitational effects can be ignored since as a design goal, underwater ROV systems use neutrally buoyant cables. Simulating the behaviour of the cable attached to an ROV can be helpful in several situations such as (i) developing control algorithms for avoiding cable tangling, (ii) handling vision issues when the cable is in front of the ROV's camera, (iii) estimating the configuration of the cable in different conditions and manoeuvres, and (iv) simulating complex manoeuvres for missions involving multiple cables and ROVs.

Figure 1: Visual simulation of Underwater tethered ROV

The behavior of a rope under external forces is generally modelled as a multi-rigid-body dynamic system (or chain system), by representing the rope as long chain of segments. In order to simulate a large number of segments in real-time, we need a fast and memory efficient method which need not necessarily be physically accurate. At the same time, the motion should look physically realistic, and controlled by a number of parameters which defines the cable behaviour. In addition, some of the unique characteristics of the water domain need to be parameterized and added to the simulation model. Simulation of dynamic systems in computer graphics mainly use force-based methods, where linear and rotational accelerations are computed from forces and torques. A time integration method is then used to update the velocities and positions of the object.

In contrast, geometry-based methods work directly on vertex positions, modifying them using iterative update equations. The main advantage of a position-based approach is its controllability. In force-based systems, overshooting problems associated with explicit integration schemes can be avoided. In addition, collision constraints can be handled easily and penetrations can be resolved completely by moving points to valid locations [1]. The position based simulations which were originally developed for the simulation of solids were also extended to the area of fluid dynamics [2] where it is not reasonable to simulate the interacting forces between the particles in real-time. In contrast to force based simulation, position based models can scale up and can be used today to simulate large multi-body systems like fabric or fur in real-time. While force based systems tend to be physically accurate under certain assumptions, position based systems aim to be visually realistic. For example, a set of simulation parameters may not correspond to real physical values, but may generate realistic object behavior as the output. The stiffness parameter implemented in the Unreal Engine which we will later discuss is an example of a parameter which influences the dynamic behaviour of the position based cable model but has no meaningful physical value (like mass or length).

Our proposed simulation method in terms of number of particles lies between the forced based system and the position based system. In simulating a long cable with a large number of segments, we aim for realtime performance by applying constraints and also use position based models. In our underwater rope implementation, we take the existing Verlet integration suggested by Jackobson [3] and modify it to simulate a realistic underwater cable.

1.1 Our Contribution

The existing rope simulation in Unreal Engine performs in a convincing way only in a light density environment like air but looked quite non-realistic in the aquatic medium. This motivated us to return to the original assumptions of the simulated implementation and modify it in order to create visually convincing underwater rope/cable simulation. In our work, we added extensive damping and random displacements to the particles and experimentally analysed and verified the resulting behaviour. Specifically, we were interested in the behaviour of variable length long cables attached to a robot at one end and to a spool in the other end as can be seen in video [4]. That kind of cables have some unique physical characteristics that could not be addressed by the current features in existing simulation frameworks. We demonstrated the behaviour of the rope as part of a larger underwater ROV simulation using Unreal Engine 4.

This paper is organized as follows: Section 2 describes the related work done on underwater cable simulation in the mechanical engineering domain and the position based simulation work done on cables in the computer graphics domain. Section 3 describes in detail the theory of position based methods with focus on cable simulation in addition to our proposed model. Section 4 describes the software we used and developed for the purpose of this work. Section 5 analyzes the results of the cable simulation. Section 6 concludes the paper with a summary of the important concepts and results presented and also outlines future work.

2 RELATED WORK

Most of the underwater cable modelling was done in the mechanical engineering domain using force based methods. Cable and chain models in general are simulated using a segment based model [5]. Buckham [6] used force based methods to simulate a cable model for use in low-tension dynamics simulation. He presented a computationally efficient and novel third-order finite element technique that provides a representation of both the bending and torsional effects and accelerates the convergence of the model at relatively large element sizes. In his paper, he managed to reduce the number of state variables defining the cubic elements of the more conventional finite element approaches . Other water cable related work reported in literature involved towed cable systems. High tension systems like towed system are quite common and a lot of work has been done on that topic. Wang investigated in his paper the parameters influencing the manoeuvre of towed cable system dynamics [7]. Lambert created a model for the dynamics and control of towed underwater vehicle systems [8]. Gonzalez created a simulation of cable pay-out and reel-in with towed fishing gears [9]. Ablow simulated the behaviour of a long cable pulled in a circular pattern [10]. Some work has been done to model the bending and the stiffness of underwater cable systems [11] [12]. Most of the simulation in the mechanical domain were designed to meet specific purpose or requirement and to serve as a guide for cable system design. For the variable length case, Prabhakar [13] developed a dynamic simulation of variable length tether in a tethered underwater vehicle system.

Position based methods are used widely in the computer graphics domain. Jackobson [3] described in detail the position based model for cable simulation. His work was the basis for the current implementation in today's game engines [14]. Our work is based on the survey paper by Bender [1] on different position based methods currently used in computer graphics.

3 ALGORITHM OVERVIEW

The Unreal game engine uses the Verlet integration method for rigid multibody simulation presented by

 ISBN 978-80-86943-41-1

Varlet [15]. The heart of the existing rope simulation is a particle system. each particle has two main variables: Its position x and its velocity v. The new position $x_{t+\Delta_t}$ and velocity $v_{t+\Delta_t}$ are computed by applying the rules:

$$x_{t+\Delta_t} = x_t + v_t\Delta_t \quad (1)$$

$$v_{t+\Delta_t} = v_t + a_t\Delta_t \quad (2)$$

where Δ_t is the time step and a_t is the acceleration. For obtaining a velocity-less representation of the above scheme, instead of storing each particle's position and velocity, we store its current position x and its previous position $x_{t-\Delta_t}$. Keeping the time step Δ_t fixed, the update rule (or integration step) is then:

$$x_{t+\Delta_t} = 2x_t - x_{t-\Delta_t} + a_t\Delta_t^2 \quad (3)$$

$$x_{t-\Delta_t} = x_t \quad (4)$$

$$x_t = x_{t+\Delta_t} \quad (5)$$

Jackobson [3] suggested in his paper that by changing the update rule to $x_{t+\Delta_t} = 1.99x_t - 0.99x_{t-\Delta_t} + a\Delta_t^2$, a small amount of drag can also be introduced to the system. This is a useful equation that can be further modified to add large drag or damping to a system in an aquatic environment. In our implementation, we added small random displacements for creating micro current effects suitable for ocean-like environment. Those micro currents prevent the rope from looking frozen in space where there are no other forces presented to the simulation. This is usually the case in long low tension cable characterized by tethered systems. Our proposed final model can be summarized by the following equations:

$$x_{t+\Delta_t} = x_t + (x_t - x_{t-\Delta_t})D_r + a\Delta_t^2 + r \quad (6)$$

$$x_{t-\Delta_t} = x_t \quad (7)$$

$$x_t = x_{t+\Delta_t} \quad (8)$$

where D_r is the drag coefficient with maximal value of 1 (no drag). It is set to 0.9 to introduce a large amount of drag typical of aquatic systems and is multiplied by the velocity term $(x_t - x_{t-\Delta_t})$ When the time step Δ_t is set equal to 1 for simulation purposes. r is the added random displacements to simulate random forces generated by underwater micro-currents. r was uniformly distributed and limits were chosen to be small enough so the random behaviour will only cause long-term effect on the cable.

The next step of the rope simulation is to apply the distance constraint. This means that the distance between adjacent particles should be kept constant. This process is done iteratively by pushing the particles directly away from each other or by pulling them closer to maintain the required distance. The following pseudo-code (Figure 2) describes this process:

```
SolveDistanceConstraint(PosA,PosB,
    TargetDistance):
 Delta = PosB-PosA
 ErrorFactor=(|Delta| - TargetDistance)/
               |Delta|
 PosA += ErrorFactor/2 * Delta
 PosB -= ErrorFactor/2 * Delta

SolveConstraints():
 for iter=0 to SolverIterations
  for ParticleIndex=0 to NumOfParticles-1
   SolveDistanceConstraint(
    Particles[i],Particle[i+1],TargetDistance
        )
  for ParticleIndex=0 to NumOfParticles-2
   SolveDistanceConstraint(
    Particles[i],Particle[i+2],2*
        TargetDistance)
```

Figure 2: Distance constraint algorithm

This pseudo-code above shows how distant constraints are implemented in Unreal Engine 4. We can see that the "SolveConstraints" function has one outer loop which is responsible to perform iterations to enforce the constraint. More solver iterations will give more stiffness to the cable. In Figure 4, the length of the rope is changed when introducing a force at one of its ends and changes in the overall length is dependent on the number of constraints and iterations applied to the rope particles. The second inner loop reduces the flexibility of the rope by enforcing constraints between particles that are separated by one other particle. That specific constraint limits the ability of the rope to bend. Figure 3 shows the difference between the two constraints. The bending constraints were useful in smoothening out the effects of random displacements added to the underwater simulation. Finally, our final solution was implemented as a new underwater rope plugin.

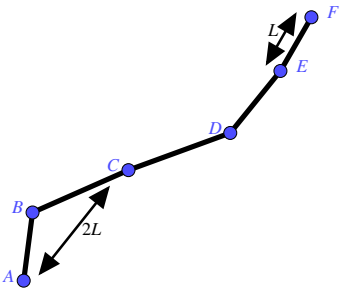


Figure 3: Length constraints and bending constraints. We can see at this diagram an example to the constraints apply on the cable segments. For example between adjacent points like E and F we require L distance which will create resistance to stretch and between points with one vertex between them like A and C we require $2L$ distance which will cause resistance to stretching with additional resistance to bending.

The rest of the changes to the rope characteristics were made by parameter changes to the model. Cable length

 ISBN 978-80-86943-41-1

was chosen to be 10 meters and the number of segments was chosen to be 100. This was done in order to create a large amount of short segments, required for visually realistic underwater simulation. With such models, any disturbance at one end of the cable will propagate slowly and will be damped by the surrounding water body.

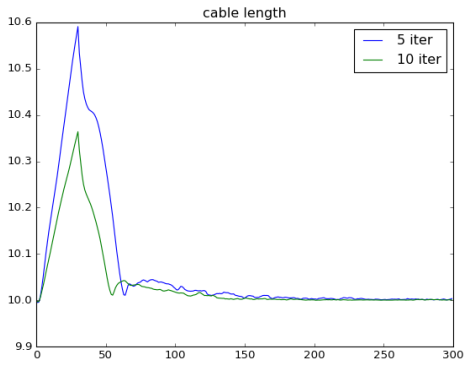


Figure 4: This figure shows the variation of the total cable length in meters with respect to the frame number. We can see that when using 10 solver iterations between frames to enforce the distance constraint of each cable segment the cable maintains its overall length more and represents a less stretchable cable.

The SolverIterations parameter (as can be seen in the pseudo code) should be chosen carefully. There is a trade-off between the stiffness or the ability to stretch and the damping mechanism introduced earlier. Since the damping is done in the Verlet stage, the constraint mechanism can still freely move all the rope particles, and due to that trade-off we limited the number of iterations. An improvement can be made to add some damping effect also in the constraint stage. That will allow more control on the cable length.

Setting the gravity to be zero was done to simulate the effect of neutral buoyancy. Usually, tethered systems are designed to meet the goal of neutral buoyancy to eliminate pulling forces from the cable in the case of non-neutral buoyancy. Additionally, the negative buoyancy of a tethered system can cause the cable to be tangled with objects on the surface of the seafloor.

Drag coefficient was chosen to be 0.9 and this value is much lower than the maximal value 1. This was done to introduce intense damping and to reduce the propagation along the cable. The random displacements coefficient was chosen empirically to be 0.1 and can be adjusted to different sea conditions. All the parameters of the simulation included the added parameters (the damping and the random displacements) can be controlled by the outside environment (like the game engine editor) and can be adapted to different types of cables with different characteristics.

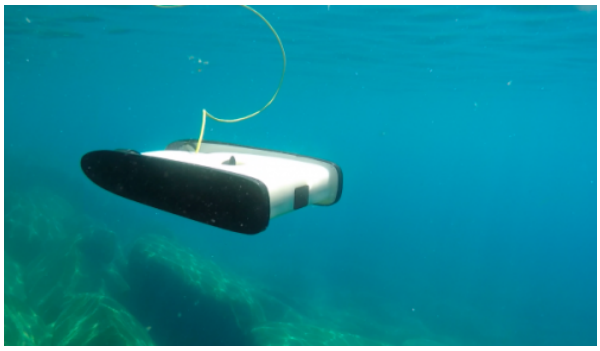


Figure 5: A small underwater OpenROV robot connected through a thin cable for video and control transmissions [16].

In our tethered ROV simulation, we have also made additional assumptions that there are no forces or relatively small forces introduced by the rope which effect the ROV position. In some cases, it makes sense for example if the mass of the ROV is relatively much higher then the mass of the rope. For example the OpenROV 5 [16] robot uses a very thin cable which handles only communication (not power) and in this case, we can assume that unless the cable is fully extended the relative force applied by the cable is relatively small. In practice, This means that the rope is not limiting the ROV movement.

3.1 Variable Length Cables

Underwater simulation of ROVs will also require modelling of cables connected to a spool that are released or retracted according to a naive logic that whenever there is a tension in the cable the cable is released. In the following, we outline a method to extend our model to a generate a variable length cable.

A flag is associated with each vertex of the cable model, and it represents whether the vertex is free to move according to the Verlet integration and the constraint mechanisms. By default, both ends of a cable will be flagged as non-free and the rest are free, since the cable is attached to both ends. In the case of a spooled cable, all the particles of the rope that are currently not released are flagged as non-free particles.

Since our model is position based, whenever the first segment from the spool side is stretched enough, typically by 10 percent of the total length, we will release a particle/vertex. After that, the Verlet integration and the constraint mechanism will move into action and will adjust the particles accordingly. In video [4] we can see that when the robot is moved the cable is pulled as necessary to maintain low tension.

We have made further modifications in the model, particularly in the areas closer to end points. Random forces were not be applied on the first free segment, to

ISBN 978-80-86943-41-1

avoid the spontaneous release of the cable due to random change of the length of the first free segment.

The spooled cable extension is done with the intention to lay a simulated foundation for the development and testing of managed tethered system. The simulation can report in real-time the current length of the cable and the estimated tension of the cable at each point along the cable. Specifically, in the beginning and the end of the cable wherein a real system we can place tension sensors as an input to the controller of the tethered system. The tension can be measured as a function of the distance between every two particles.

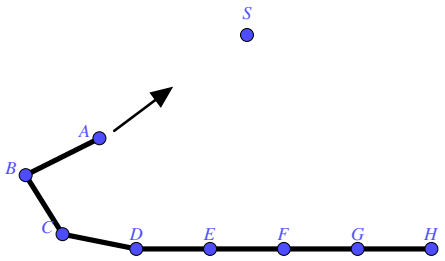


Figure 8: A 2D model of a flexible cable where one of the end points A is moved with a constant velocity v towards a target S

4 METHODS AND TOOLS

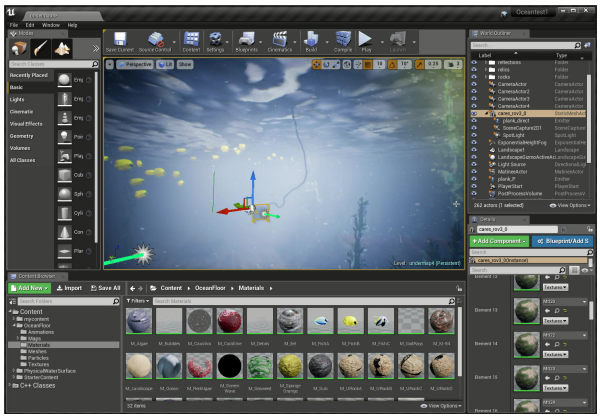


Figure 6: Unreal Engine 4 editor environment.

Figure 7: Experimental verification of motion of an extensible cable. We colored the cable in a checkers like pattern to enable the extension of the cable to be observable.

The main aim of this work has been to generate a convincing and realistic behaviour of an underwater tethered robot using the simulation framework provided by the Unreal Engine 4 (Figure 6). A live video demo can be seen in videos [17] [4] and in Figure 7. The experiments were done in the editor environment (not as a packed game). The robot seen in those figures was moved manually while the cable was attached to both ends. The new plugin is maintained and can be downloaded from here [18].

In addition, to have finer control over the simulation, an additional 2D simulation was created to demonstrate the proposed method. We used the Jupyter [19] python notebook environment to generate the output seen in figures 9 and 11. The 2D simulation is maintained under the following link [20]. Figure 8 illustrates the cable configuration used in the 2D simulation.

5 EXPERIMENTAL RESULTS

We created a simple 2D computer simulation to simulate the effects of moving one edge of the cable while the other end is pinned (Figure 7). Figure 9 shows the behaviour of the rope with and without damping with respect to time. The wave motion continues to propagate through the rope when there is no damping whereas with damping the wave energy slowly decays and random forces are becoming more dominant. In Figure 10 we can see our desired effect when using coefficient 0.9. The movement of the cable at one end does not affect the other end, so long as there is no tension in the cable segments.

Figure 11 shows the the random forces effect. In this experiment we look at the cable configuration in the 2D space after the system is stabilized ($t >> 0$) with and without random forces. We can see that the random forces create a kind of memory loss effect of the shape of the cable. This effect is important when there are no other significant forces (or they are close to zero) in the system. When we don't add the random force, the cable tends to stand still in contrast to what would be expected in a dynamic aquatic environment.

ISBN 978-80-86943-41-1

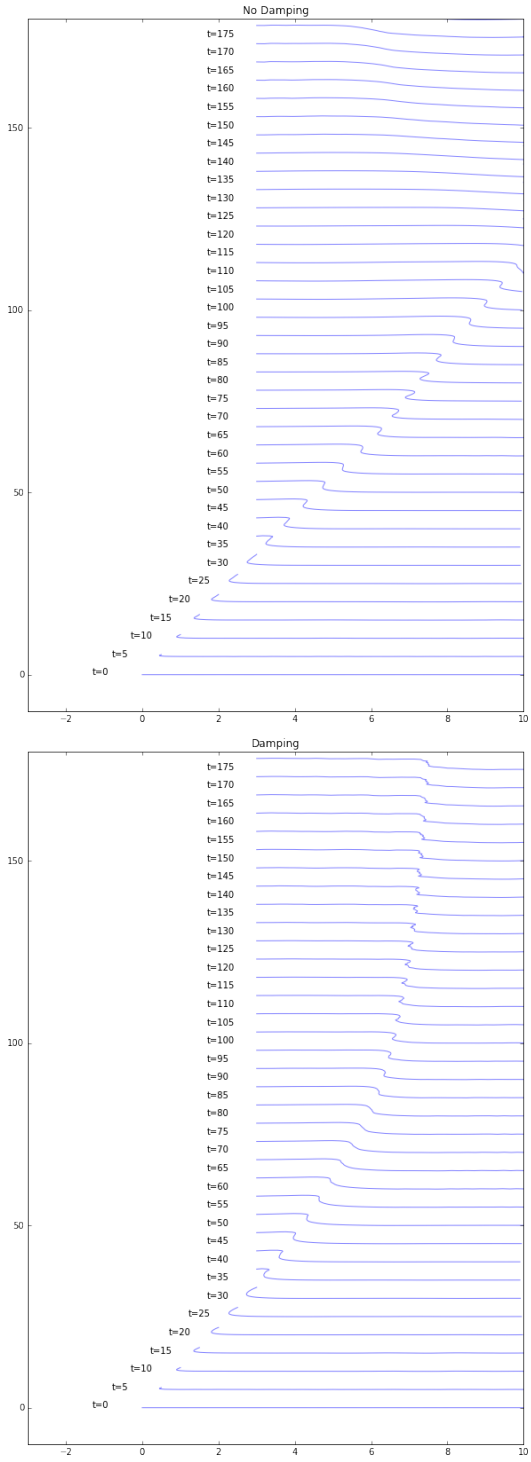


Figure 9: The damping effect. This figure shows the cable in different times. In t=0 we start to move the edge of the cable in the direction up and right along the "xy" plane. The first figure shows the results without damping and the second shows the behaviour of that cable with damping coefficient of 0.98. We can clearly see that the damping is absorbing the wave energy as we would expect in aquatic systems.

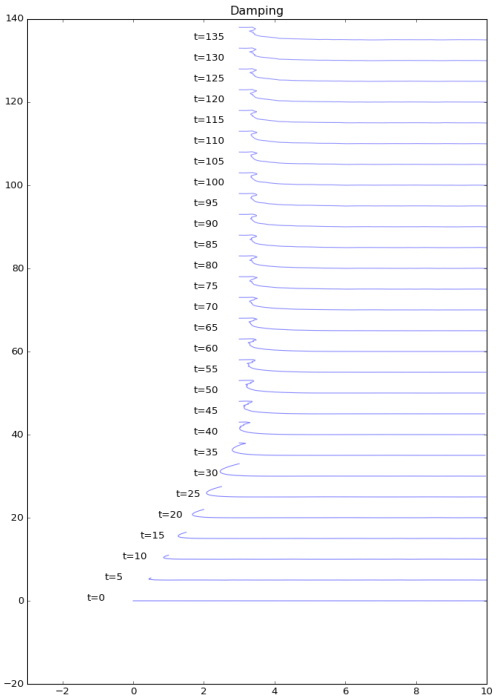


Figure 10: Damping with 0.9 coefficient. Tunning the coefficient to 0.9 causes the desired effect for underwater simulation in the case of a low tension cable in an underwater environment. Disturbance on one side remains local.

6 CONCLUSIONS AND FURTHER RESEARCH

In terms of performance, the modifications to the current model didn't require more computational effort. In fact, if we assume neutral buoyancy of the cable we can remove the gravitational forces from the simulation to reduce computational time. This can be useful in cases where a large number of cable/rope like object are simulated. Generally speaking, we can say that a cable is a linear 3d object curve which can be efficiently computed by modern CPUs.

Using the position based approach allowed us to easily modify the current model by adding drag and random displacements and in the future to apply other constraints for inter-rope tangling and ROV interactions. Further research will also deal with the forces applied to and by the cable to the objects that it is connected to. This can be done by measuring the length of the segments as described in section 3.1. Currently, we assume that there are no forces and torques applied by the rope which effect the ROV movement, and so we can improve future simulation by adding those forces to the simulation. Additionally, we added simple ran-

ISBN 978-80-86943-41-1

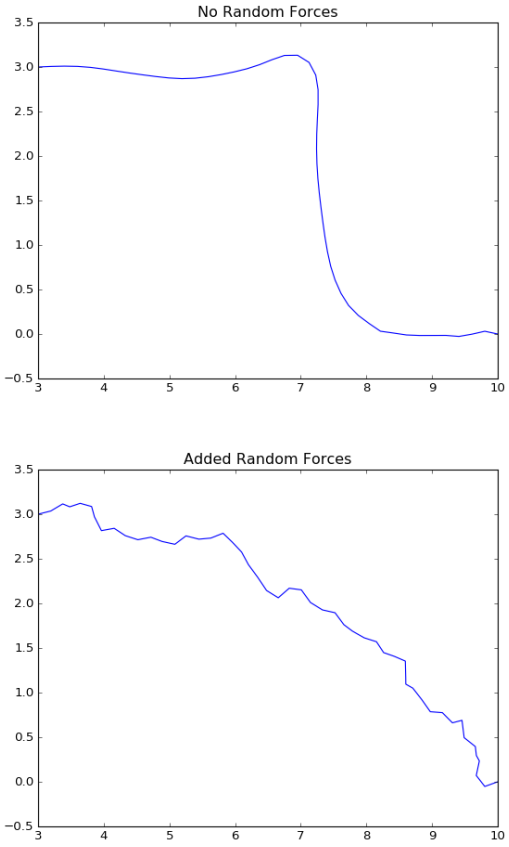


Figure 11: Introducing random forces to the system. Both images show the cable state after the system is stabilized (t»0). The first and the second images show the cable state with and without random forces respectively. We can see the "Memory Loss" that we would expect to see in a marine-like environment with underwater currents.

dom displacements with even normal distribution for all the segments. In real environments that is usually not the case and the currents are influencing the cable differently for each segment. Underwater currents are more similar to air turbulence and do not contain high frequency changes.

In our research, we found this simulation to be particularly useful in cases were the robots sees its own cable as presented in Figure 12. This fact may disrupt computer vision algorithms - especially those based on tracking using landmark features from the images and assume that these landmarks are not moving in the scene. With this new type of simulation, that kind of behaviour can be simulated in a manner close to real cable behaviour. New computer vision algorithms can be developed to mitigate that behaviour and new control algorithms can be developed to manage the cable configuration.

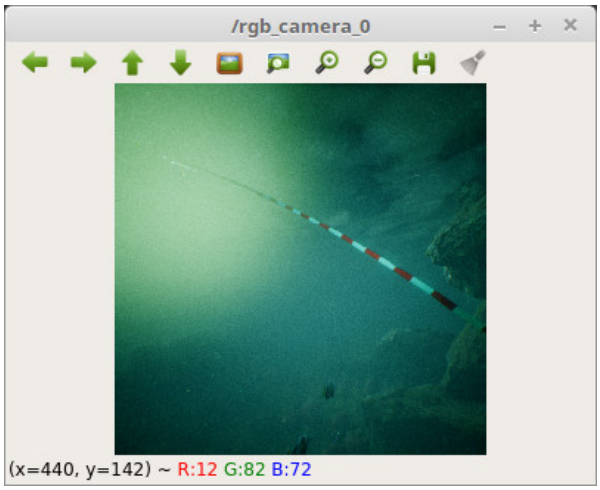


Figure 12: An underwater robot's camera view of its own tether cable

In this paper, we demonstrated a visually convincing underwater cable simulation. The current state of the art model implemented in the latest version of the Unreal Engine 4 was thoroughly investigated and the needed modifications to the model for underwater simulation were described in detail. We presented a novel approach to the underwater simulation and the unique characteristics of such a medium. We showed results in a 2d computer simulation for finer analysis of the simulation results. The simulation is robust and controlled by a large number of parameters as previously described. Finally, the modified model was implemented in Unreal engine 4 as a new underwater cable component available for download with provided demos demonstrating the new cable behaviour [18].

7 REFERENCES

[1] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin, "A survey on position-based simulation methods in computer graphics," in *Computer graphics forum*, vol. 33, no. 6. Wiley Online Library, 2014, pp. 228–251.

[2] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 104, 2013.

[3] T. Jakobsen, "Advanced character physics," in *Game Developers Conference*, vol. 3, 2001.

[4] Underwater cable reel simulation video. https://youtu.be/DO-x2RaZHso.

[5] R. Marshall, R. Jensen, and G. Wood, "A general newtonian simulation of an n-segment open chain model," *Journal of Biomechanics*, vol. 18, no. 5, pp. 359–367, 1985.

[6] B. Buckham, F. R. Driscoll, and M. Nahon, "Development of a finite element cable model for use in low-tension dynamics simulation," *Journal of*

ISBN 978-80-86943-41-1

Applied Mechanics, vol. 71, no. 4, pp. 476–485, 2004.

[7] Z. Wang and G. Sun, "Parameters influence on maneuvered towed cable system dynamics," *Applied Ocean Research*, vol. 49, pp. 27–41, 2015.

[8] C. Lambert, M. Nahon, B. Buckham, and M. Seto, "Dynamics and control of towed underwater vehicle system, part ii: model validation and turn maneuver optimization," *Ocean engineering*, vol. 30, no. 4, pp. 471–485, 2003.

[9] F. González, A. de la Prada, A. Luaces, and M. González, "Real-time simulation of cable payout and reel-in with towed fishing gears," *Ocean Engineering*, vol. 131, pp. 295–307, 2017.

[10] C. Ablow and S. Schechter, "Numerical simulation of undersea cable dynamics," *Ocean engineering*, vol. 10, no. 6, pp. 443–457, 1983.

[11] J. Burgess *et al.*, "Bending stiffness in a simulation of undersea cable deployment," *International Journal of Offshore and Polar Engineering*, vol. 3, no. 03, 1993.

[12] J. Gobat and M. Grosenbaugh, "Time-domain numerical simulation of ocean cable structures," *Ocean Engineering*, vol. 33, no. 10, pp. 1373–1400, 2006.

[13] S. Prabhakar and B. Buckham, "Dynamics modeling and control of a variable length remotely operated vehicle tether," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1255–1262.

[14] Cable component in unreal engine 4. https://docs.unrealengine.com/latest/INT/Engine/Components/Rendering/CableComponent/.

[15] L. Verlet, "Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules," *Physical review*, vol. 159, no. 1, p. 98, 1967.

[16] Openrov. https://www.openrov.com/.

[17] Cable simulation video. https://youtu.be/_QoMUSlQCsg.

[18] Cable sim project. https://github.com/UnderwaterROV/UWCableComponent.

[19] Jupyter. http://jupyter.org/.

[20] Cable sim notebook. https://github.com/UnderwaterROV/underwaterrov/blob/master/notebooks/rope.ipynb.

 ISBN 978-80-86943-41-1

3D RECONSTRUCTION OF CORONARY ARTERIES FROM ROTATIONAL X-RAY ANGIOGRAPHY

Chaima Oueslati, Sabra Mabrouk, Faouzi Ghorbel
National School of Computer Science
CRISTAL laboratory
GRIFT research group
Manouba 2010
Tunisia

Mohamed Hedi Bedoui
Faculty of Medecine of Monastir
TIM Team
Laboratory of Biophysics
Monastir 5019
Tunisia

ABSTRACT

X-ray angiography has been an effective modality for diagnosing coronary artery disease representing one of the leading causes of death. A 3D reconstruction of the coronary arteries is a very important step to facilitate the interpretation of angiograms. In this paper, we propose a 3D reconstruction method of the coronary arteries. In order to improve the pairwise matching performance of our approach, we introduce artificial interest points (nodes) to respect the topology variation between 3D vascular trees. New similarity measures are proposed to take into consideration the non-rigid coronary artery movement. To measure the performance of the proposed method, we evaluate the proposed method on clinical data. Results show that accuracy of vessel centerlines has the average projection error equal to 0.5 mm for 28 different patients.

Keywords

X-Ray angiography, 3D reconstruction, 3D vascular trees, artificial nodes

1 INTRODUCTION

Coronary heart disease has been for decades one of the primary dangers to human health essentially due to coronary atherosclerosis. In clinical practice, several image acquisition techniques were developed for cardiac examination, the most currently used one is the X-ray angiography thanks to the temporal and spatial resolutions of the provided images and the speed of the medical examination associated with it. However, the 2D nature of produced images makes it difficult to interpret the overlaying and crossing structures.

The sequence of projections thus available can be exploited to perform a 3D reconstruction of the coronary tree that, associated with 3D visualization tools and quantification, will be able to bring a substantial help in documentation of injuries and search for optimal incidences of observation in which the interventional cardiologist can perform the angioplasty procedure. This 3D reconstruction, however, represents a real challenge because of structures movement.

Among the rich literature on coronary artery reconstruction [1], two main categories have been proposed model-based methods (modelling) and tomographic reconstruction.

The first inspired by the principles of computer vision exploits a modeling technique by proceeding the mapping of extracted 2D entities from 2 to 4 projections [2], This method try to find a binary representation of the 3D/4D structure of the coronary arteries. In order to perform the reconstruction, authors in [7] compute the 2D vessel centerlines from the projections and then search correspondences in them to represent the motion without using the ECG signal.

Authors in [8] use an optimisation scheme from two uncalibrated angiographic images with 14 parameters included by minimizing two errors distances and direction vectors of corresponding skeleton points in projections, then they use the epipolar constraints and the vessel surface is extracted fitting elliptical cross-sections to carry out the correspondence of centerline points between the two projections.

While the second categorie proceeds by static tomographic reconstruction from a weak number of projections. In fact, tomographic reconstruction methods aim at reconstructing 3D / 4D volume of attenuation coefficients [3].
The forward projections are computed using the 3D reconstructions of markers on the guidewire in [9] and the ECG-gated tomographic reconstruction in [10]. In some cases, the forward projected images are prepared

to extract some features (e.g. centrelines) for the registration [10].

With the movement related to the breathing of the patient and the heart, the branches undergo additional nonlinear deformations and they are different for each patient. Two similar branches from one phase to another do not have the same curvature or the same coordinates. This problem makes it difficult to map arteries for each phase. These variations can be due to various difficulties related to acquired 2D images (presence of noise, reduction of contrast medium, structures superposition, etc.), the movement of the heart, the breathing of the patient or to differences related to the angle of view. In [5], authors describe an inaccurate tree matching algorithm for recording 3D trees of non-isomorphous coronary arteries over time.

In this paper, a 3D reconstruction method of coronary arteries from two projections of monoplanes, uncalibrated and acquired under two different projection angles is described. Based on geometric characteristics and the characteristics of the nodes, the similarity between nodes is calculated using new criteria. A step which takes into account the variation of topology between 2D vascular trees is presented and propose to insert artificial interest points. In our case, the skeleton matches are not found point by point using the epipolar constraints, but branch to branch using the similarity measure between the extremity nodes of two branch in the 2D sequence.

The rest of this paper is organized as follows. In section 2, the proposed method is described in detail. In section 3, experimental results are presented and conclusions are draw in the last section.

2 PROPOSED METHOD

From the 2D skeletons of the segmented image, the 3D structure of the vascular tree can be extracted. In our case, the reconstruction is done on a pair of projections. Based on the information of the central line, a 2D tree is deduced for each segmented image [4]. Then, each remarkable point such as the bifurcation point and the end point is considered as a point of interest. From one projection to another, a branch can actually disappear, appear or be of different length depending on the heart phase.

The framework of the proposed method is shown in Fig.1.

2.1 Matching approach

2.1.1 The similarity measure

We proposed two metrics to determine the similarity of the interest points in the two projections. The new node-to-node metrics allow taking into account non-rigid movements of the coronary arteries.

We note that instead of choosing a system of absolute coordinates linked to the acquisition room, we prefer a relative presentation of the one projection according to the other, as a simplification, the first image is considered as a reference, the passage of one view to another is reacted by a rigid transformation composed of a 3 * 3 rotation matrix R and a 3 * 1 translation matrix T.

Similarity measures significantly influence pairwise matching results. The choice of the node-to-node metrics is important, moreover, between two coronary trees representing no completely successive phases of the cardiac cycle, the movement could be important therefore these metrics need to be as efficient as possible.

The Euclidean distance between the coordinates of two nodes in different trees can be used since the coronary trees are extracted from the same acquisition of the same patient.
The first metric is the Euclidean distance between a node after applying the rigid transformation in the first image and a node in the second image. In fact, we want to determine the closest node to the second node after applying the rigid transformation.

To compute the second metric, we first test if the two nodes have the same type (the two nodes are bifurcation points or endpoints), and if they have the same number of neighbors of type bifurcation and endpoint. If the two points verify these two conditions, two distances are calculated, the first is the sum of the Euclidean distances between the bifurcation neighbors and the second is the sum of the Euclidean distances between the ends neighbors.

The similarity measure represents the minimum pondered sum of the first metric and the second metric. We note that based on the choice of constraints and their measure threshold value, we obtain the best matching result which is only composed of a pair of tree similar nodes.

2.1.2 Insertion of artificial nodes

In the proposed method, we consider that the matching accuracy is more important than the corresponding node number. In order to refine the matching process by adding artificial nodes to take into account the topology variation between the vascular trees, we apply the Artificial nodes insertion algorithm [6].

These artificial nodes correspond to bifurcation points and leaf nodes that only exist in one of the vascular trees.

Figure 2 shows two examples of artificial nodes insertion. We treat the artificial node insertion in two cases:

 ISBN 978-80-86943-41-1

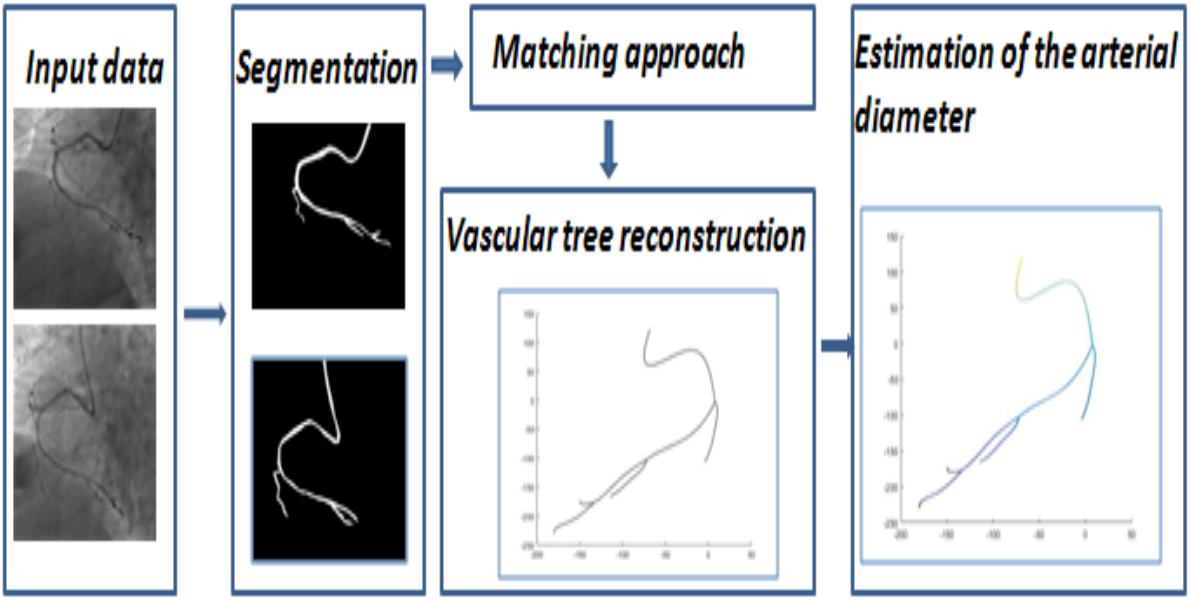


Figure 1: The framework of the proposed method

the artificial end nodes insertion in the first row and artificial bifurcation node insertion in the second row.

2.2 Vascular tree Reconstruction

After exhaustive merging, we are left with a set of three-dimensional points that reflects the underlying anatomy, every two connected three-dimensional points represent a 3D branch. As we mentioned in the previous section, two similar branches from one phase to another do not have the same curvature, the same coordinates, and the same length.

In order to respect the topology of the arterial tree, an additional step is added making correspondence between particular points of each similar branches.

2.2.1 Matching two branches' points

We intend to make the branch-length reparameterization and extract the same number of points from each branch. For each point p_i, its correspondent point p_j is the closest one in the other branch after applying the rigid transformation.

Given the matched nodes between centerline points in the two projections, three-dimensional reconstruction is deduced straight-forward using triangulation [12].

Once the 3D centerline established, we estimate the artery diameter with the method proposed in [4].

3 EXPERIMENTAL RESULT

We evaluate the proposed method on clinical data provided by Mongi-Slim Hospital of La Marsa Tunisia. The datasets consist of 28 patients having each 8 to 12 images. The metric used for the evaluation of the proposed method is the average of the distances between the initial and reconstructed skeletons. This metric can be seen as the difference value between back projection results and original image pixels. it represents the sum of distance error between all back projection nodes X^P of image and the original nodes X:

$$e = \sum_{i=1}^{N} (X_i^P - X_i)^T (X_i^P - X_i) \quad (1)$$

Where e is the re-projection error and N is the total number of nodes.

The optimization procedure is performed by the Levenberg-Marquardt algorithm [11]. The idea is to decompose the objective function in the form of quadratic substitution functions, which makes the minimization simple and "parallelizable" from a computer point of view.// The average error of projection was reduced from 2.8 $mm \pm 1.55$ before the optimization to 0.5 $mm \pm 0.011$ after(Fig 3).

We note that the threshold associated with the similarity measure is empirical chosen and identical for each correspondence test. To determine the best configuration, we check if all the nodes pairs that need to be matched are selected.

The program execution is optimized thanks to the recursion of the mapping method which makes the execution time very fast.

ISBN 978-80-86943-41-1

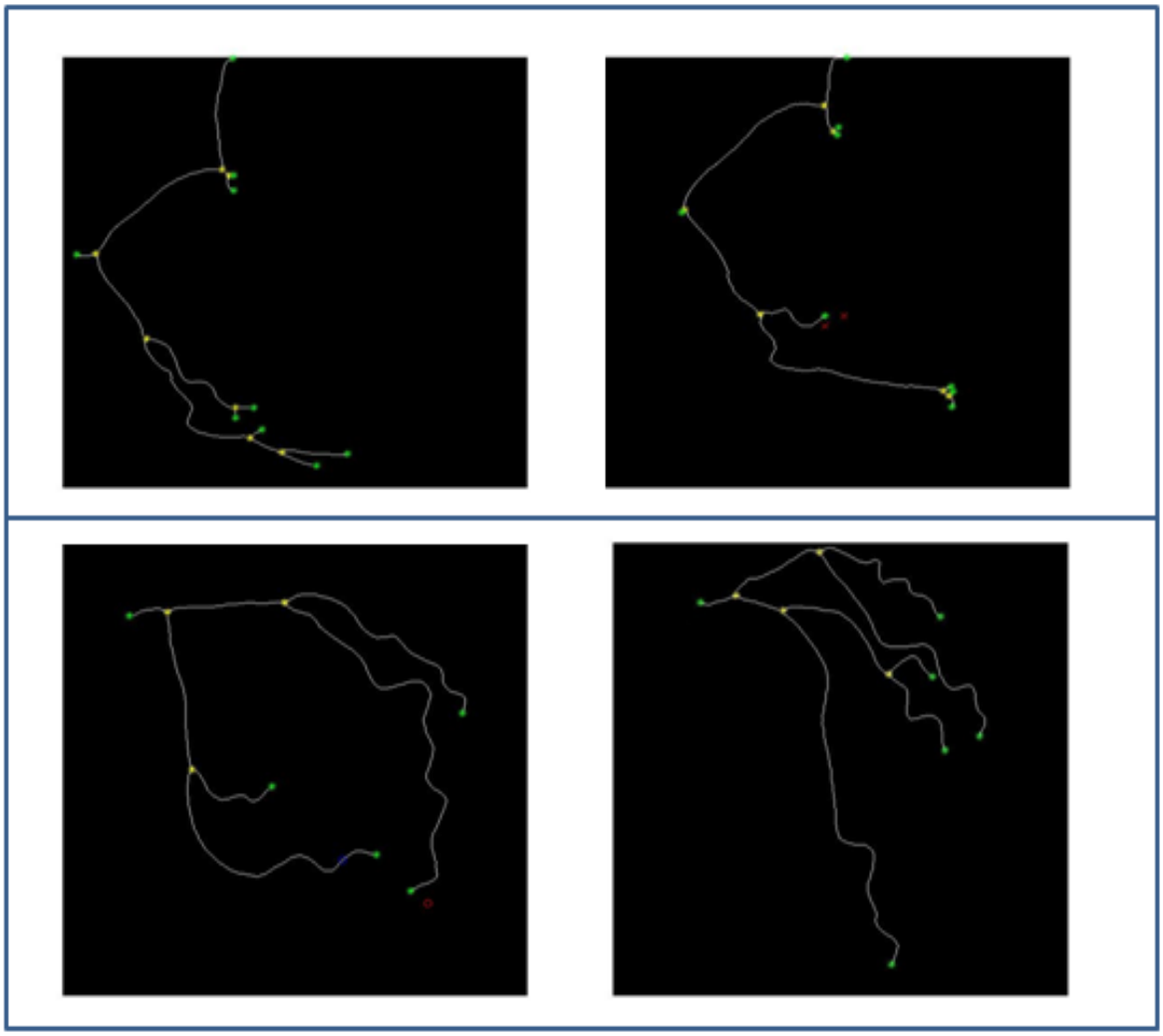


Figure 2: Insertion of artificial nodes: First row: insertion of artificial end points (red cross), second row: insertion of bifurcation node (blue circle) and end point (red circle)

4 CONCLUSION

In this paper, we proposed a 3D reconstruction method of the coronary tree from two projections of monoplanes, uncalibrated and acquired under two different projection angles. The geometric characteristics and the characteristics of the nodes are used to evaluate the similarity between the nodes and the edges using two new criteria. To take into account the variation of topology between coronary trees due to their extraction, a step which consists of adding an artificial node to our coronary trees is introduced.

5 REFERENCES

[1] ÇIMEN, Serkan, GOOYA, Ali, GRASS, Michael, et al. Reconstruction of coronary arteries from X-ray angiography: A review. Medical image analysis, 2016, vol. 32, p. 46-68.

[2] CHEN, S.-YJ et CARROLL, John D. Kinematic and deformation analysis of 4-D coronary arterial trees reconstructed from cine angiograms. IEEE transactions on medical imaging, 2003, vol. 22, no 6, p. 710-721.

[3] SCHOONENBERG, Gert, FLORENT, Raoul, LELONG, Pierre, et al. Projection-based motion compensation and reconstruction of coronary segments and cardiac implantable devices using rotational X-ray angiography. Medical image analysis, 2009, vol. 13, no 5, p. 785-792.

[4] MABROUK, S., OUESLATI, C., et GHORBEL, F. Multiscale Graph Cuts Based Method for Coronary Artery Segmentation in Angiograms. IRBM, 2017, vol. 38, no 3, p. 167-175.

[5] FEUILLÂTRE, H., NUNES, J.-C., et TOUMOULIN, C. An improved graph matching algorithm for the spatio-temporal matching of

ISBN 978-80-86943-41-1

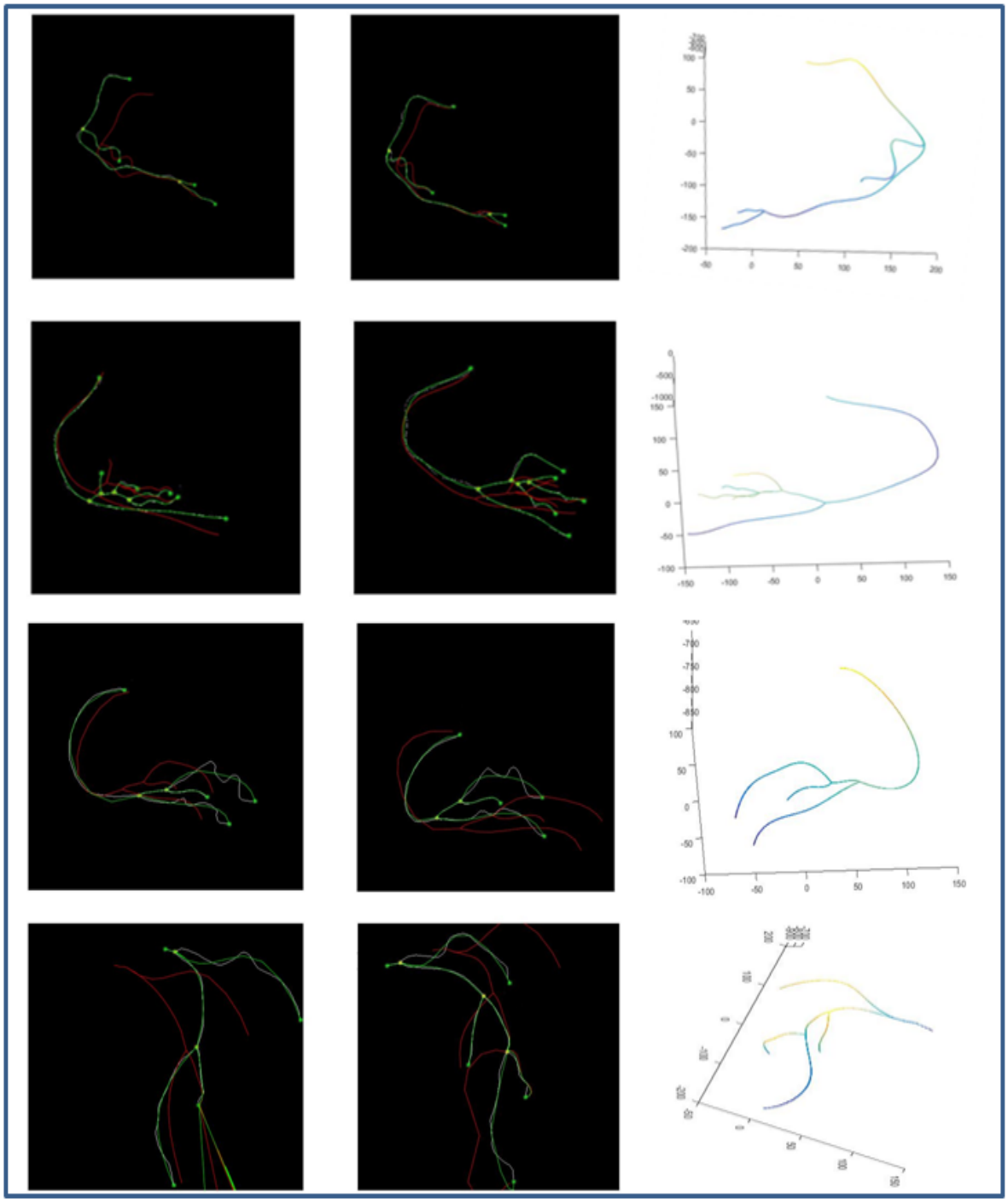


Figure 3: Reprojections of reconstruction results: (column 1 et 2): Erroneous, reprojected,and ground truth center-lines are overlapped in red, green and white respectively. (column 3) 3D reconstructed arteries tree

ISBN 978-80-86943-41-1

a coronary artery 3D tree sequence. IRBM, 2015, vol. 36, no 6, p. 329-334.

[6] FEUILLÂTRE, Hélène, NUNES, Jean-Claude, et TOUMOULIN, Christine. Inexact coronary tree matching algorithm with artificial nodes. In : Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016. p. 4153-4156.

[7] BLONDEL, Christophe, MALANDAIN, Grègoire, VAILLANT, RÃ©gis, et al. Reconstruction of coronary arteries from a single rotational X-ray projection sequence. IEEE Transactions on Medical Imaging, 2006, vol. 25, no 5, p. 653-663.

[8] YANG, Jian, WANG, Yongtian, LIU, Yue, et al. Novel approach for 3-D reconstruction of coronary arteries from two uncalibrated angiographic images. IEEE Transactions on Image Processing, 2009, vol. 18, no 7, p. 1563-1572.

[9] MOVASSAGHI, Babak, RASCHE, Volker, FLORENT, R., et al. 3D coronary reconstruction from calibrated motion-compensated 2D projections. In : International Congress Series. Elsevier, 2003. p. 1079-1084.

[10] HANSIS, Eberhard, SCHÄFER, D., DÖSSEL, O., et al. Projection-based motion compensation for gated coronary artery reconstruction from rotational x-ray angiograms. Physics in medicine and biology, 2008, vol. 53, no 14, p. 3807.

[11] MORÉ, Jorge J. The Levenberg-Marquardt algorithm: implementation and theory. In : Numerical analysis. Springer, Berlin, Heidelberg, 1978. p. 105-116.

[12] Hartley R, Zisserman A. Multiple View Geometry in Computer Vision. Cambridge: Cambridge university press; 2003.

 ISBN 978-80-86943-41-1

Performance evaluation of face alignment algorithms on "in-the-wild" selfies

Ivan Babanin
Moscow Institute of Physics and Technology, Adorable Inc.
Department of Innovations and High Technology
Institutskiy Pereulok, 9
Russian Federation, 141701, Moscow region, Dolgoprudny
ivan.babanin@phystech.edu

Aleksandr Mashrabov
Moscow Institute of Physics and Technology, Adorable Inc.
Department of Innovations and High Technology
Institutskiy Pereulok, 9
Russian Federation, 141701, Moscow region, Dolgoprudny
mashrabov@phystech.edu

ABSTRACT

Recently mobile apps, which beautify human face or apply cute masks to a human face, become very popular and gain lots of attention in media. These tasks require very precise landmarks localization to avoid "uncanny valley" effect. We introduce the new dataset of selfies, that were taken on mobile devices, and robustly evaluate and compare different state-of-the-art approaches to the task of face alignment. Evidently, our dataset allows to reliably rank face alignment algorithms that is superior to the most popular dataset in that area of research.

Keywords

Benchmark testing, Face, Shape, Machine learning, Robust measurement, Mobile devices, Face alignment

1 INTRODUCTION

The problem of face detection and face alignment has been the focus in computer vision for more than two decades. Recently many research teams have focused on the collection and the annotation of real-world datasets of facial images captured in-the-wild. Such datasets evolve into challenges and encourage many scientists to develop face alignment algorithms that are robust to different pose variations. Although, latest challenges focus on 3D alignment and robust face alignment in a video, although the diversity of datasets with precise annotations for semi-frontal faces is low. However, this case is trendy since people use phones more often than desktop for social media and search on the Internet. This entails the rise of social platforms focused on images messages like Instagram and Snapchat and tools that beautify photos like Snapchat lenses, FaceTune. Another common case that requires exact face alignment is virtual makeup tools. Such applications like Youcam Makeup with more than 100M downloads help to find how you would look if your lips are colored by pink lipstick, if shadows under eyes are green, etc.

We present the first selfies dataset carefully annotated them with 68 fiducial points in a manual manner according to current labeling standards. Our goal is the creation of small dataset to robustly compare state-of-the-art academic and commercial approaches. Also, we check the correlation between overall face alignment quality and quality of tracking key points in specific face areas (mouth, eyes, contour). Example of face annotation is depicted on Figure 1.

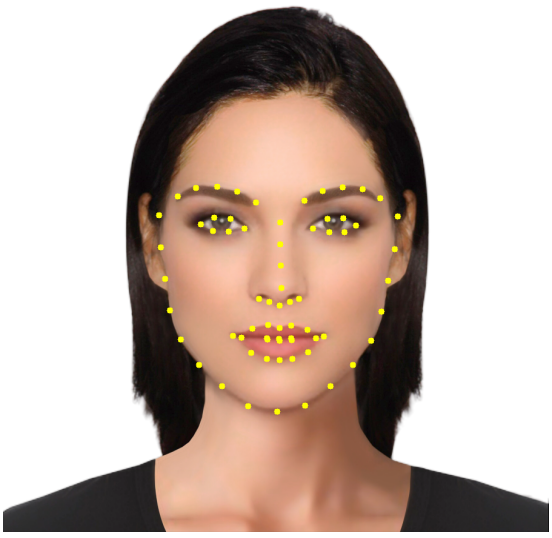


Figure 1: Annotated image with 68 key points [1]

2 REVIEW OF EXISTING DATASETS

2.1 Datasets of annotated images

Labeled Face Parts in the Wild (LFPW) database [1] contains 1287 images downloaded from google.com, yahoo.com, and flickr.com. The dataset covers a broad range of appearance variation, including pose, lighting, expression, occlusion, and individual differences. The provided ground truth consists of 35 landmark points.

Helen database [2] contains 2330 images in good resolution downloaded from the flickr.com website. Annotation with 194 landmarks is very precise, but most images are taken not on a mobile phone.

The Annotated Faces in-the-wild (AFW) [3] database which consists of 250 images with 468 faces. Six facial landmark points for each face are provided.

Menpo Challenge database [4] consists of 300W [5] train and test data, iBug dataset. Overall it has 5658 annotated semi-frontal and 1906 annotated profile facial images. Semi-frontal images are provided with 68 landmarks and profile with 39 landmarks. Recently, competition on face alignment was organized on that dataset in July 2017 at top-tier computer vision conference CVPR 2017.

These are the most widely used publicly available databases of images with fiducial points annotation. Although Menpo and Helen databases have enough key points in annotation, original photos in those datasets mostly aren't selfies.

3 RECENT SOLUTIONS

3.1 State-of-the-art academic approaches

W. Wu: Method in [6] used a deep network (VGG-16 and Resnet-18) to regress to a parametric form of the shape of multiple datasets and another network to make the final decision. It showed incredible results in Menpo Challenge 2017 [4] with the 2-nd place and almost real-time performance. The code is not available online; we privately asked authors to evaluate their algorithm on our dataset.

M. Kowalski: Method in [7] used a VGG-based alignment network to correct similarity transforms, extracting features from the entire face images rather than patches around facial key points, and then a fully-convolutional network that finally localizes 68 key points. The code with the pretrained model is available online.

Z. He (Zhenliang): Method in [8] used already known FEC-CNN architecture as a basic method for facial landmark detection with a bounding box invariant algorithm that reduces the prediction sensitivity to face detector and model ensemble technique that is adapted for further performance improvement. The code is not available online; we privately asked authors to evaluate their algorithm on our dataset.

X.-H. Shao: Method in [9] used a sub-network of VGG-19 for landmark heatmap and affinity field prediction at the former stage, and Pose Splitting Layer that regresses basic landmarks at a latter stage. According to its pose, each canonical state is distributed to the corresponding branch of the shape regression sub-networks for the whole landmark detection. The code is not available online; we privately asked authors to evaluate their algorithm on our dataset.

A. Bulat: Method in [10] used a stack of 4 "Hourglass Networks" for landmark localization with a residual block, trained on a very large yet synthetically expanded 2D facial landmark dataset. That leads to remarkable robustness to initialization of parameters and yaw angle of images. The code is open-sourced with pre-trained models.

G. Tzimiropoulos: Method in [13] was implemented in [12] and used parametric linear models of both shape and appearance of an object, typically modeled PCA. The AAM objective function involves the Gauss-Newton minimization of the appearance reconstruction error concerning the shape parameters.

G. Trigeorgis: Method in [17] used a combined and jointly trained convolutional recurrent neural network architecture of cascaded regressors that allows the training of an end-to-end to alleviate problems of existing approaches such as not coherent training process of regressors, the prevalence of handcrafted features. The recurrent module facilitates the joint optimization of the regressors by assuming the cascades are forming a nonlinear dynamical system, in effect, fully utilizing the information between all cascade levels by introducing a memory unit that shares information across all levels. The code is open-sourced.

3.2 Proprietory production systems

Dlib: A very popular fast face alignment library that is widely used as a baseline. It used an ensemble of regression trees under the hood and came with a pretrained model for 68 facial key points localization. It is open-sourced library and is available at http://dlib.net/.

iOS face alignment: Apple Vision framework that came live with iOS11 in September 2017 provides 65 landmarks. Due to the inconsistency of localization of key points, we compared the accuracy of key points localization only related to mouth region.

4 PROPOSED SOLUTIONS

There are many existing benchmarks for face alignment algorithms, but our goal was to collect a relatively small

[1] http://www.bbc.co.uk/newsbeat/article/32115303/mr-and-mrs-perfect-in-the-real-world

 ISBN 978-80-86943-41-1

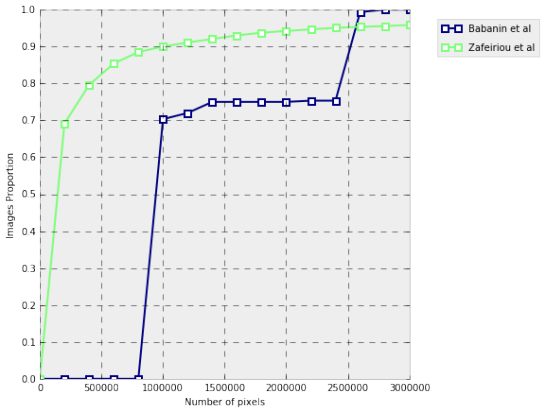


Figure 2: Cumulative distribution of pixels in images

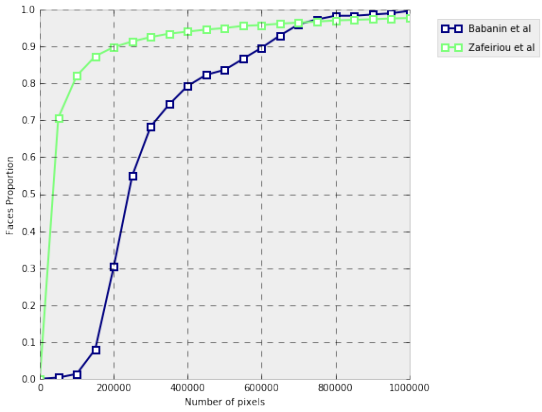


Figure 3: Cumulative distribution of pixels in face rectangles

set of photos that adequately characterize the diversity of selfies. Such images are relatively "easy" compared to almost profile face images [4], so the quality of labeling becomes crucial to make reasonable conclusions. Hence we filtered all photos with an occluded face (by arm, scarf, etc.) and filtered very dark selfies since many popular tasks like face beautification don't make sense in such case.

The number of selfies, that passed initial half-automated filtration, exceeds 5000 images. At last stage, our goal was to ensure the diversity of identities (no more than four photos from each person) that uniformly cover the full range of emotions. At this point, we used 3D Face Morphable Models [15] to fit each image to estimate albedo and shape coefficients. Albedo coefficients describe the identity of a person, helping to limit the number of photos from each person very precisely. Set of shape coefficients describe the full range of emotions [19]; therefore, we applied Principal Components Analysis algorithm [16] to this set to select the photos that demonstrate the diversity of emotions in real-life. Whereas, we used open-source library 4dface [18] to fit each image to 3D face model. 4dface framework operates with local features rather than rough pixel values that results in much more robust fitting against variations in images conditions. The final dataset contains only 300 photos, that allows to compute final metrics very quickly.

We collected dataset of selfies taken on mobile phones by users of the mobile application on behalf of Adorable Inc. All photos were taken on frontal camera and had a resolution at least 720*1080 that is bigger than the majority of images in Menpo dataset [4]. More specifically, 69 percents of photos in Menpo have a resolution less than 200,000 pixels. Thus the majority of pictures in current popular benchmarks is four times smaller than images in our dataset (see Figure 2).

Furthermore, we compared the area of face rectangles in our dataset and Menpo dataset (see Figure 3). 70 percents of face rectangles in Menpo dataset [4] has the area less than 50,000 pixels, although 70 percents of face rectangles in our dataset have the area more than 200,000 pixels.

5 EVALUATION METRICS

In the biggest competitions on face alignment main metric for evaluation is the point-to-point Euclidean distance normalized by the interocular distance [5]. However, as noted in [3], this error metric doesn't provide robust results for profile faces with small interocular distance. Hence, we propose two types of normalizations. In particular, we used the Normalized Mean Error (Normalized Point-to-Point error) defined as:

$$\text{NME} = \frac{1}{N}\sum_{i=1}^{N}\frac{|gt_i - pr_i|_2}{d}, \qquad (1)$$

where gt denotes the ground truth landmarks for a given face, pr - the corresponding prediction. And d is:

The diagonal ground truth bounding box [11], computed as $d = \sqrt{w_{facebbox}^2 + h_{facebbox}^2}$. This normalization is standard.

The square-root (geometric) of the ground truth bounding box of corresponding face region, computed as $d = \sqrt{w_{bbox} * h_{bbox}}$. This new type of normalization depends on characteristic values of that particular region (e.g. size of mouth is much smaller than size of face).

Moreover, our goal is to compare alignment of different face regions: mouth (N=20 points), eyes and brows (N=22 points), contour (N=17 points), so that we have six error values to compare aforementioned face alignment approaches.

However, as noted in [4] mean errors without corresponding standard deviations are not reliable metrics to compare approaches and to make reasonable conclusions. Therefore, we provide our evaluation in the form of cumulative error distribution (CED) curves. After that we find the area-under-the-curve (AUC) taking

ISBN 978-80-86943-41-1

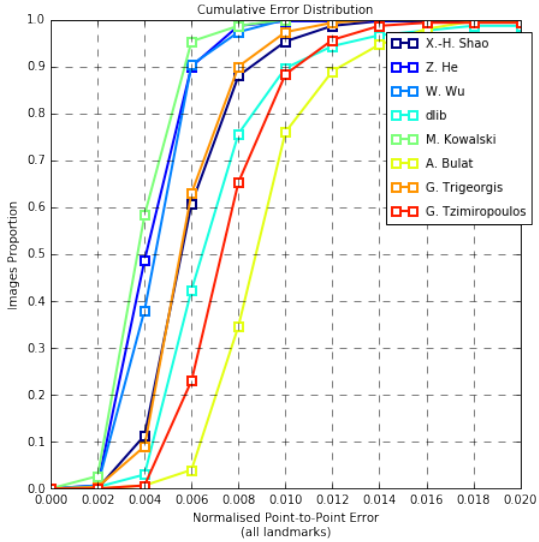


Figure 4: CED curve for entire face, diagonal normalization

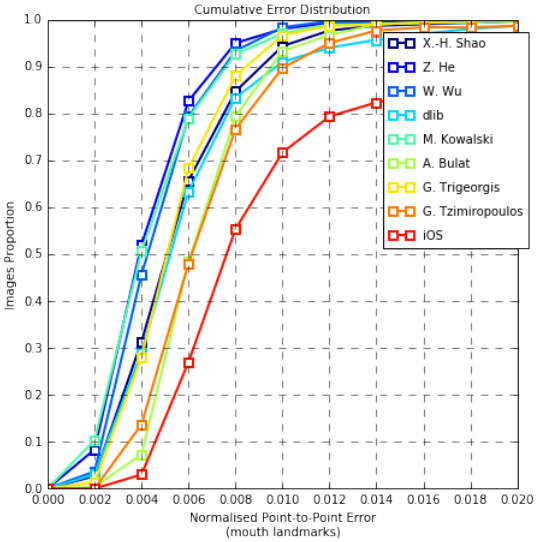


Figure 6: CED curve for mouth region, diagonal normalization

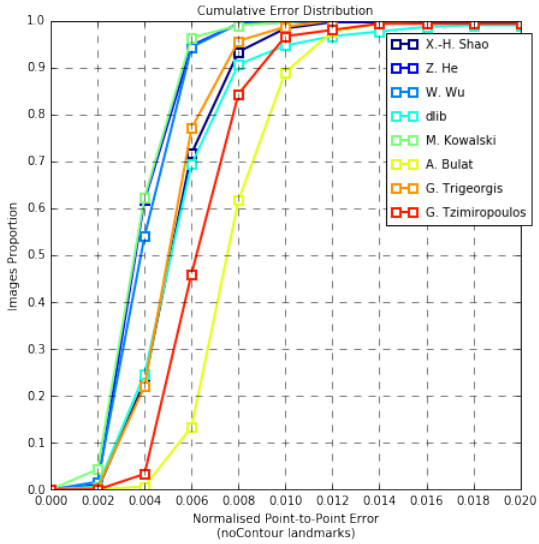


Figure 5: CED curve for entire face without contour, diagonal normalization

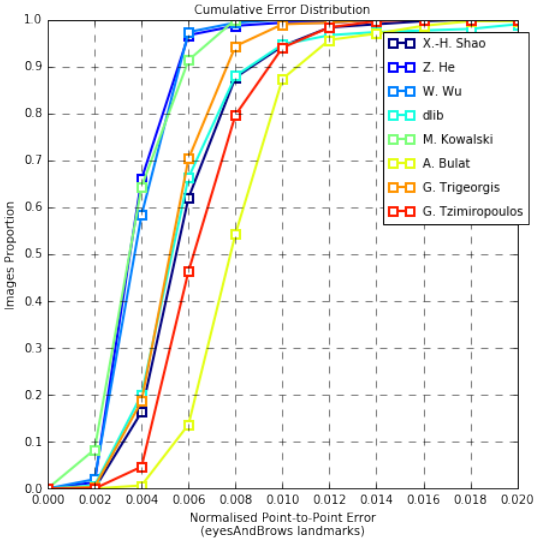


Figure 7: CED curve for eyes and brows region, diagonal normalization

only those images that have the error less than 0.03 [20]. Besides, that error rate for geometric normalization and mouth, eyes and brows, no controur region is 0.30. Another important metric is the failure rate of each method that is a proportion of images with error more than 0.03, which describes very poor face alignment that cannot be used for any further face modification like digital makeup.

6 EXPERIMENTAL RESULTS

In this section we will describe key observations, validate our hypothesis and show that declared goals are achieved. Bulat et al. [10] performed much worse than other methods. Also, there is a tremendous gap between state-of-the-art methods that use complicated Deep Learning approaches and old-fashioned regression methods, decision trees methods. This observation was already noted in [4]. We compared all approaches with first types of normalization and found out that the ranking is almost the same for different face regions (Figure 4 for all landmarks, Figure 5 for all landmarks except contour, Figure 6 for mouth landmarks, Figure 7 for brows and eyes landmarks). The huge advantage of our approach compared to Menpo challenge [4] is much smaller deviation at the much smaller size (300 vs 5335). Evidently, Kowalski et al. [7] showed the best result on our dataset: deviation on our dataset is 1/3 of a mean value, but in Menpo dataset deviation is more than a mean value. Since that ranking of results in Menpo dataset is not reliable and our approach allows to compare algorithms more consistently.

ISBN 978-80-86943-41-1

	Bulat[10]	Tzim.[13]	Kow.[7]	dlib	Trig.[17]	Shao[9]	Wu[6]
Tzim.[13]	1e-28	—	—	—	—	—	—
Kow.[7]	6e-51	6e-51	—	—	—	—	—
dlib	4e-36	6e-11	6e-51	—	—	—	—
Trig.[17]	5e-50	3e-44	2e-50	1e-26	—	—	—
Shao[9]	3e-49	1e-45	6e-51	5e-22	1e-03	—	—
Wu[6]	6e-51	6e-51	7e-36	7e-51	1e-48	2e-49	—
He[8]	6e-51	6e-51	9e-20	8e-51	2e-48	4e-50	1e-11

Table 1: Wilcoxon test for all 68 keypoints with first normalization

Surprisingly, mean error and a standard deviation are very similar (Figure 4, Figure 5, Table 2, Table 4) on 68 key points (entire face) and 41 key points (without contour). Our initial hypothesis was that it is difficult to make labeling of contour landmarks consistent. Therefore we expected that error on entire face without contour region would be much less. It turned out to be false.

The only region that suits for comparing keypoint localization algorithm employed in iOS is mouth region (Figure 6, Table 6). Anyway, the quality of that algorithm is clearly very poor. In fact, failure rate of iOS algorithm is more than 10 percents, when failure rate of other algorithms is less than 1 percent. Additionally, the only method with deviation more than mean value is the iOS algorithm (Table 6, Table 7). Also, Bulat et al. [10] bypass Tzimiropoulos et al. [13].

Another region that has slightly different ranking is eyes and brows region (Figure 7, Table 8). There is almost indistinguishable difference between leader He et al. [8] and runner-up Kowalski et al. [7].

Moreover, we compared all algorithms to each other using Wilcoxon signed-rank test to assure that our method produces reliable results and allows to compare algorithms. For each image, we computed error rate on 68 key points (first type of normalization by diagonal of face rectangle) and ran the test on 300 pairs of values (see Table 1).

Another part of our research consists of comparing two types of normalization. That second type is geometric normalization by taking a square root of sides of a corresponding face region. This almost doesn't affect the ranking of all face regions, but relative deviation becomes much smaller for mouth region (Table 6, Table 7) and remains the same for other regions.

7 CONCLUSION

We achieved our goal to create a small dataset that allows to efficiently and robustly rank and differentiate current state-of-the-art face alignment approaches. From our best knowledge it is the only such dataset. Summing up, the quality of face tracking of popular proprietory systems is far worse than top-level academic approaches. The quality of method by M.Kowalski et al. [7] shows excellent results from qualitative and quantitative points.

In our dataset, the overall mean error is smaller than in [4] that is an implication of nature of photos (well lighting, not extreme head rotation poses). The important observation is that quality of key points localization of different face regions (eyes, mouth, contour) highly correlates with quality on entire face. Another significant comment is that we achieved much smaller deviation without artificial clipping of photos with large head rotations. We believe that there is still a room for research to create a relevant small dataset with accurate labeling that represents the full diversity of face poses not limited to selfies. Our goal for further research is to create openly available benchmark for 3D landmark tracking on "in-the-wild" selfies.

8 ACKNOWLEDGMENT

Thanks to Adorable Inc. for providing access to data with face images.

9 REFERENCES

[1] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, IEEE Transactions on Pattern Analysis and Machine Intelligence (T- PAMI), 35(12), 2930-2940, 2013.

[2] V. Le, J. Brandt, Z. Lin, L. Bourdev, T.S. Huang, Interactive facial feature localization, In Proceedings of European conference on computer vision (ECCV) (pp. 679-692) Springer, 2012.

[3] X. Zhu, D. Ramanan, Face Detection, Pose Estimation, and Landmark Localization in the Wild, In CVPR 2012, 2012.

[4] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, J. Shen, The Menpo Facial Landmark Localisation Challenge: A step towards the solution', In CVPRW 2017, 2017.

[5] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge, In ICCV 2013, 2013.

[6] W. Wu, S. Yang, Leveraging Intra and Inter-Dataset Variations for Robust Face Alignment,

 ISBN 978-80-86943-41-1

In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPRW), Faces-in-the-wild Workshop/Challenge, 2017.

[7] M. Kowalski, J. Naruniec, and T. Trzcinski, Deep Alignment Network: A convolutional neural network for robust face alignment, In Proceedings of the International Conference on Computer Vision Pattern Recognition (CVPRW), Faces-in-the-wild Workshop/Challenge, 2017.

[8] Z. He, J. Zhang, M. Kan, S. Shan, X. Chen, Robust FECCNN: A High Accuracy Facial Landmark Detection System, In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPRW), Faces-in-the-wild Workshop/Challenge, 2017.

[9] X.-H. Shao, J. Xing, J. Lv, C. Xiao, P. Liu, Y. Feng, C. Cheng, and F. Si, Unconstrained Face Alignment without Face Detection, In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPRW), Faces-in-the-wild Workshop/Challenge, 2017.

[10] A. Bulat, G. Tzimiropoulos, How far are we from solving the 2d and 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks), ICCV 2017, 2017.

[11] G. Chrysos, E. Antonakos. P. Snape, A. Asthana, S. Zafeiriou, A Comprehensive Performance Evaluation of Deformable Face Tracking "In-the-Wild", International Journal of Computer Vision, 2017.

[12] J. Alabort-i-Medina, E. Antonakos, J. Booth, P. Snape, S. Zafeiriou, Menpo: A comprehensive platform for parametric image alignment and visual deformable models, In Proceedings of ACM international conference on multimedia, (ACM'MM) (pp. 679-682). ACM, 2016.

[13] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, M. Pantic, Active orientation models for face alignment in-the-wild, IEEE Transactions on Information Forensics and Security, 9(12), 2024-2034, 2014.

[14] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, In IEEE proceedings of international conference on computer vision and pattern recognition (CVPR), (pp. 532-539), 2013.

[15] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, A 3D Face Model for Pose and Illumination Invariant Face Recognition, In Proceedings of the 6th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) for Security, Safety and Monitoring in Smart Environments, Genova (Italy) - September 2-4, 2009.

[16] J. Shlens, A Tutorial on Principal Component Analysis, `https://www.cs.cmu.edu/~elaw/papers/pca.pdf`, 2004.

[17] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, S. Zafeiriou, Mnemonic Descent Method: A recurrent process applied for end-to-end face alignment, Proceedings of IEEE International Conference on Computer Vision Pattern Recognition (CVPR 16), Las Vegas, NV, USA, June 2016.

[18] P. Huber, Z. Feng, W. Christmas, J. Kittler, M. Ratsch, Fitting 3D Morphable Models using Local Features, IEEE International Conference on Image Processing (ICIP 15), Quebec City, Canada, 2015.

[19] M. Yu, B. P. Tiddeman, Facial Feature Detection and Tracking with a 3D Constrained Local Model, WSCG 2010 conference proceedings, Copyright UNION Agency Science Press, pp: 181-188, WSCG 2010.

[20] H. Yang, X. Jia, C. C. Loy, P. Robinson, An Empirical Study of Recent Face Alignment Methods, arXiv preprint arXiv:1511.05049, 2015.

 ISBN 978-80-86943-41-1

	Mean	Std	Median	MAD	Max Error	$AUC_{0.03}$
M. Kowalski *et al.* [7]	0.0039	0.0012	0.0038	0.0008	0.0091	0.8706
Z. He *et al.* [8]	0.0043	0.0014	0.0040	0.0007	0.0150	0.8571
W. Wu *et al.* [6]	0.0045	0.0013	0.0043	0.0007	0.0097	0.8511
G. Trigeorgis *et al.* [17]	0.0058	0.0016	0.0055	0.0009	0.0121	0.8083
X.-H. Shao *et al.* [9]	0.0059	0.0020	0.0055	0.0012	0.0200	0.8025
dlib	0.0071	0.0032	0.0063	0.0011	0.0271	0.7647
G. Tzimiropoulos *et al.* [13]	0.0076	0.0024	0.0072	0.0012	0.0237	0.7453
A. Bulat *et al.* [10]	0.0091	0.0025	0.0086	0.0011	0.0242	0.6982

Table 2: Entire face, diagonal normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.03}$
M. Kowalski *et al.* [7]	0.0056	0.0018	0.0054	0.0011	0.0131	0.8146
Z. He *et al.* [8]	0.0061	0.0020	0.0058	0.0010	0.0214	0.7951
W. Wu *et al.* [6]	0.0064	0.0018	0.0061	0.0010	0.0139	0.7862
G. Trigeorgis *et al.* [17]	0.0082	0.0023	0.0078	0.0013	0.0171	0.7251
X.-H. Shao *et al.* [9]	0.0085	0.0028	0.0080	0.0017	0.0286	0.7168
dlib	0.0101	0.0045	0.0090	0.0016	0.0387	0.6650
G. Tzimiropoulos *et al.* [13]	0.0110	0.0034	0.0104	0.0018	0.0338	0.6349
A. Bulat *et al.* [10]	0.0130	0.0035	0.0123	0.0015	0.0346	0.5677

Table 3: Entire face, geometric normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.03}$
M. Kowalski *et al.* [7]	0.0038	0.0013	0.0037	0.0008	0.0104	0.8739
Z. He *et al.* [8]	0.0039	0.0014	0.0037	0.0006	0.0180	0.8699
W. Wu *et al.* [6]	0.0040	0.0012	0.0039	0.0007	0.0109	0.8650
G. Trigeorgis *et al.* [17]	0.0051	0.0014	0.0049	0.0009	0.0110	0.8301
X.-H. Shao *et al.* [9]	0.0053	0.0019	0.0050	0.0011	0.0214	0.8225
dlib	0.0057	0.0031	0.0051	0.0011	0.0304	0.8113
G. Tzimiropoulos *et al.* [13]	0.0064	0.0021	0.0061	0.0011	0.0237	0.7859
A. Bulat *et al.* [10]	0.0078	0.0018	0.0076	0.0011	0.0156	0.7414

Table 4: Entire face without contour, diagonal normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.30}$
M. Kowalski *et al.* [7]	0.0140	0.0041	0.0136	0.0022	0.0320	0.9534
Z. He *et al.* [8]	0.0146	0.0044	0.0140	0.0019	0.0619	0.9515
W. Wu *et al.* [6]	0.0151	0.0034	0.0149	0.0019	0.0376	0.9497
G. Trigeorgis *et al.* [17]	0.0191	0.0046	0.0184	0.0024	0.0405	0.9365
X.-H. Shao *et al.* [9]	0.0199	0.0064	0.0188	0.0033	0.0737	0.9336
dlib	0.0210	0.0102	0.0189	0.0033	0.1066	0.9300
G. Tzimiropoulos *et al.* [13]	0.0240	0.0066	0.0231	0.0031	0.0808	0.9202
A. Bulat *et al.* [10]	0.0290	0.0053	0.0281	0.0027	0.0667	0.9034

Table 5: Entire face without contour, geometric normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.03}$
Z. He *et al.* [8]	0.0043	0.0023	0.0039	0.0012	0.0224	0.8566
M. Kowalski *et al.* [7]	0.0045	0.0026	0.0039	0.0014	0.0204	0.8512
W. Wu *et al.* [6]	0.0046	0.0020	0.0042	0.0012	0.0154	0.8468
G. Trigeorgis *et al.* [17]	0.0054	0.0024	0.0050	0.0013	0.0193	0.8204
X.-H. Shao *et al.* [9]	0.0055	0.0028	0.0049	0.0014	0.0253	0.8156
dlib	0.0060	0.0037	0.0052	0.0016	0.0278	0.7988
A. Bulat *et al.* [10]	0.0065	0.0023	0.0061	0.0012	0.0213	0.7831
G. Tzimiropoulos *et al.* [13]	0.0068	0.0037	0.0061	0.0015	0.0412	0.7748
iOS	0.0193	0.0682	0.0074	0.0020	0.6440	0.6572

Table 6: Mouth landmark region, diagonal normalization

ISBN 978-80-86943-41-1

	Mean	Std	Median	MAD	Max Error	$AUC_{0.30}$
Z. He *et al.* [8]	0.0550	0.0214	0.0522	0.0123	0.1984	0.8167
M. Kowalski *et al.* [7]	0.0565	0.0244	0.0545	0.0150	0.1869	0.8118
W. Wu *et al.* [6]	0.0591	0.0193	0.0571	0.0125	0.1367	0.8029
G. Trigeorgis *et al.* [17]	0.0696	0.0232	0.0671	0.0128	0.1627	0.7679
X.-H. Shao *et al.* [9]	0.0711	0.0268	0.0665	0.0145	0.2243	0.7631
dlib	0.0769	0.0362	0.0678	0.0149	0.2573	0.7437
A. Bulat *et al.* [10]	0.0845	0.0209	0.0824	0.0116	0.2503	0.7185
G. Tzimiropoulos *et al.* [13]	0.0879	0.0428	0.0825	0.0127	0.6046	0.7103
iOS	0.2575	1.0055	0.0989	0.0232	12.5684	0.5756

Table 7: Mouth landmark region, geometric normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.03}$
Z. He *et al.* [8]	0.0038	0.0012	0.0037	0.0006	0.0125	0.8734
M. Kowalski *et al.* [7]	0.0038	0.0014	0.0036	0.0008	0.0078	0.8731
W. Wu *et al.* [6]	0.0039	0.0011	0.0037	0.0007	0.0095	0.8698
G. Trigeorgis *et al.* [17]	0.0053	0.0016	0.0051	0.0010	0.0132	0.8230
X.-H. Shao *et al.* [9]	0.0059	0.0021	0.0055	0.0012	0.0174	0.8041
dlib	0.0059	0.0040	0.0052	0.0011	0.0484	0.8039
G. Tzimiropoulos *et al.* [13]	0.0066	0.0020	0.0062	0.0011	0.0145	0.7816
A. Bulat *et al.* [10]	0.0081	0.0023	0.0078	0.0013	0.0256	0.7307

Table 8: Eyes and Brows landmark region, diagonal normalization

	Mean	Std	Median	MAD	Max Error	$AUC_{0.30}$
M. Kowalski *et al.* [7]	0.0244	0.0075	0.0237	0.0052	0.0509	0.9188
Z. He *et al.* [8]	0.0246	0.0071	0.0240	0.0036	0.0793	0.9181
W. Wu *et al.* [6]	0.0251	0.0059	0.0247	0.0039	0.0531	0.9162
G. Trigeorgis *et al.* [17]	0.0343	0.0091	0.0328	0.0050	0.0862	0.8856
X.-H. Shao *et al.* [9]	0.0379	0.0123	0.0354	0.0065	0.1103	0.8738
dlib	0.0383	0.0241	0.0333	0.0066	0.2949	0.8725
G. Tzimiropoulos *et al.* [13]	0.0421	0.0099	0.0407	0.0056	0.0856	0.8596
A. Bulat *et al.* [10]	0.0522	0.0129	0.0504	0.0057	0.1676	0.8261

Table 9: Eyes and Brows landmark region, geometric normalization

ISBN 978-80-86943-41-1

A New Robust and Blind Image Watermarking Scheme In Frequency Domain Based On Optimal Blocks Selection

Nesrine Tarhouni
REsearch Groups in Intelligent Machines, National Engineering School of Sfax
Sfax,3038,Tunisia
nesrine.tarhouni@enis.tn

Maha Charfeddine
REsearch Groups in Intelligent Machines, National Engineering School of Sfax
Sfax,3038,Tunisia
maha.charfeddine@enis.tn

Chokri Ben Amar
REsearch Groups in Intelligent Machines, National Engineering School of Sfax
Sfax,3038,Tunisia
chokri.benamar@enis.tn

ABSTRACT

Image, audio and video are the first media affected by hacking due to the availability of the internet and to the high speed connection. One of the solutions to solve such problems is watermarking. Digital watermarking is the process of embedding an imperceptible and a robust signature into a digital signal. In this paper, we focus on image watermarking. We have embedded the watermark in the frequency domain using Discrete Cosine Transform. The choice of the blocks where we insert the watermark bits depends on a preprocessing study on the original and compressed-decompressed image. Then we have implemented a blind detection algorithm. We tried to enhance the security of our technique by applying an Arnold transform to the embedded watermark. Finally, we have tested the robustness of our method by applying many attacks to the watermarked images using Stirmark 3.1. The results demonstrate that our method yields a high level of imperceptibility and robustness against JPEG compression, unique and double Stirmark attacks.

Keywords

Image watermarking, Discrete Cosine Transform, Arnold.

1 INTRODUCTION

The appearance of digital data is a recent revolution in the world of signal processing. Indeed, switching from analogue to digital has made handling more convenient. The transmission is faster, the storage more economical, the indexing more efficient and the copying easier.

Certainly, simplifying the access to the identical copy has facilitated hacking. Image, audio and video are the first media affected by this serious problem,as such data can be tampered and used without authorization,can be copied with preserving the image quality and with unlimited number of copies.

Several researchers have tried different methods to prevent or at least slow down the copying of these multimedia data. For example, steganography, which aims to hide a message into a data in such a way that an eavesdropper cannot detect the presence of the message [Pooj15]. Also, the cryptography, which attempts to protect the content by making it unreadable and the output is inexplicable without the knowledge of the key [Sur17]. Besides, the digital watermarking is the technique of hiding information into a digital content (image, audio, video, etc) to protect it from dishonest manipulations. In contrast to steganography and cryptography, the watermarked data are available, exploitable and the presence of the hidden message is known [Man16].

Digital watermarking was proposed for many applications: Tracing traitors, content authentication, indexing and event detection from video sequences and essentially for copyright protection [Fch15] [Nam17] [Bou11] [Ali10].

The techniques of digital watermarking have particular characteristics regarding to the given application. Therefore, the watermarking systems don't follow the same properties. Generally, there are three important features that are usually treated in the most applications:

Imperceptibility: This property is related to the insertion scheme where the embedded watermark not degrade the media quality. The watermarks do not create visible artifacts in images and also don't alter the bit rate of the video or introduce audible noises in audio signals[Wke15].

Robustness: It means the resistance of the digital

watermarking technique against changes made to the watermarked media. It depends on the aplication, if the watermarking is used for copyright protection, then the watermark has to be available after different modifications. The watermarks should not get destroyed as a result of unintentional or malevolent distortions like cropping, resampling, rotation, scaling and compression. On the other hand, if it is used for content authentication, the watermarks get disappeared whenever the content is modified so that the loss of integrity of document must be detected [Hai13].

Capacity: The capacity of insertion represents the quantity of information inserted in the content. More the capacity is low more the imperceptibility and robustness are relevant [Son16].

The digital watermarking scheme consists of two steps: the embedding process and the detection process. The techniques according to the embedding domain are classified into two categories : insertion domain without transformation and with transformation where we hide the watermark in the frequency domain [Mas10] or the multiresolution one [Mel11]. An example for the first class, in [Sra17] which proposed a watermarking approach in spatial domain based on LSB. Color watermark is composed of three different binary watermarks. The composite color watermark is embedded by substituting the least significant bit of the intensity values of the cover image. Its detection scheme is blind. For the second class, [Moh16] focused on image watermarking in YCoCg-R color space. In the proposed method, DCT is applied to Y of the host image and each bit of watermarks is embedded in three different blocks. Also, Arnold transformation is used to scramble Y and the watermark.The authors in [Lam15], proposed a semi blind approach based on DCT and linear interpolation to protect and authenticate the source such as quran text image. The RGB image is transformed to YUV then, applying DCT and quantification to each 8*8 block of the original image and the watermark. Finally, generating the watermarked image by applying the linear interpolation equation. The watermark could be detected in most cases under various types of attacks when the parameter t of the equation was set near to one. PSNR value is over than 34 dB while SSIM, VIF and UQI values are close to 1, and the NQM exceeds 30 dB. [Ssh14] proposed a method for authentication and copyright protection, the authors applied DWT and SVD to the low frequency subband LL of both cover and watermark images. Then, they embedded the singular values of the watermark image in singular values of the host image. The detection scheme uses the cover image to extract the watermark.
In this paper, a blind and robust image watermarking scheme resistant against many different types of attacks such geometric distortions, common signal processing and JPEG compression.

The main contributions include:
1. The spatial domain based on LSB provides low degradation of image quality and important capacity but it is not robust.Hence, we decided to substitute the watermark in the LSB but in the frequency domain.
2. The choice of the suitable blocks to insert the watermark bits depends on a preprocessing study on the original and compressed-decompressed image.
3. The watermark is scrambled using Arnold transformation to ameliorate the security level.
4. The detection of the watermark is directly from the attacked image.
5. Our method resists to double attacks of Stirmark.

2 PRELIMINARIES

2.1 YCbCr color space

YCbCr is the well known space used for video and digital photography system, where Y is the luminance component, Cb and Cr are respectively the blue-difference and red-difference components. The transformation from RGB space to YCbCr is in the equation (1) and the conversion from YCbCr to RGB is done by using the formula (2) [Sub17].

$$\begin{cases} Y = 0.2989 \times R + 0.5866 \times G + 0.1145 \times B \\ Cb = -0.1688 \times R - 0.3312 \times G + 0.5 \times B \\ Cr = 0.5000 \times R - 0.1181 \times G - 0.0816 \times B \end{cases} \quad (1)$$

$$\begin{cases} R = 1.0 \times Y + 0.0 \times Cb + 1.403 \times Cr \\ G = -0.1688 \times Y - 0.3312 \times G + 0.5 \times B \\ B = 0.5000 \times Y - 0.1181 \times G - 0.0816 \times B \end{cases} \quad (2)$$

2.2 DCT and IDCT transforms

The DCT transform is used in this work, in order to convert the original signal from spatial domain to frequency domain. For the original image f(x, y), DCT transform could be shown as follows:

$$F(u,v) = c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \cos \frac{\pi u(2x+1)}{2M} \cos \frac{\pi v(2y+1)}{2N} \quad (3)$$

Here M and N are the rows and columns. The c(u) and c (v) could be shown as follows:

$$c(u), c(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } u, v = 0 \\ 1 & \text{si } otherwise \end{cases} \quad (4)$$

The inverse of DCT is IDCT,is used to convert the signal from the frequency domain to the spatial domain.

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u) \times c(v) F(u,v) \times \cos[\frac{\pi}{M} u(x+\frac{1}{2}] \cos[\frac{\pi}{N} v(y+\frac{1}{2}] \quad (5)$$

ISBN 978-80-86943-41-1

With images, most of the energy prevails in low frequency. While the high frequency can be neglected as it results little visible distortion such as JPEG compression.

2.3 Arnold transform

One of the purposes of our method is to improve the security level by scrambling the watermark. One such scrambling techniques is the Arnold transformation. The specificity of this transformation is that the image will not be affected after certain iterations. The number of iteration is called also 'Arnold period'. The Arnold Transform of the image is as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \quad (6)$$

3 THE PROPOSED IMAGE WATERMARKING SCHEME

The proposed image watermarking technique consists of three parts: the embedding process, the application of Stirmark attacks and extracting the watermark. These parts are depicted in Figure 1.

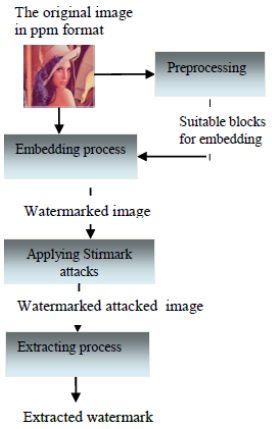


Figure 1: The flow chart of the watermarking approach

3.1 DCT-LSB-Arnold watermark embedding algorithm

In this section, we clarify in details the inserting steps of the watermark and these steps are described in figure 3.

Step1: Reading the watermark, transform it to binary image and scramble it by Arnold transform eight times. This number is related to the image size and based on experimental results.

Step2: Loading the host RGB image.

Step3: Converting the RGB image to YCbCr color space. YCbCr is chosen because its components offer minimum correlations [Aro15]. The YCbCr color space is used in the JPEG standard, therefore, the use of this color space allows us to improve the robustness results.

Step4: Separating the Y, Cb and Cr components and Choosing the 'Y' component and elicit matrix whose size is M*M, Y is chosen because it is tolerant to the wellknown attack JPEG compression

Step5: Splitting Y into 8*8 blocks so that we have a total of (M*M)/64 blocks.

Step6: All pixels of the blocks are substracted by 128 then transformed into frequency domain using DCT transform, then quantized and scanned by applying zigzag to all 64 DCT coefficients.

Step7: Selecting the middle band frequency coefficients to embed the watermark for the following reasons: Embedding in the low frequency will affect the visual quality of the image because the most energy is situated there. so, the requirement for imperceptibility will not be reached. Besides, the high band frequency is the most easily removed region after applying lossy compression, low pass filter and image noise. So choosing that band won't meet the robustness requirement. Therefore, middle band frequency is chosen to embed the mark as long as it usually provides good imperceptibility and robustness results in different watermarking schemes with several data (audio, images) [Mah12].

The previous steps are inspired from the algorithm of the JPEG standard. We adopt this idea to assure that our watermarking technique will be robust to JPEG compression attack directly without operating a decompression phase before the extraction process is triggered, which makes our approach original comparing with others [Job17].

Step8: Studying and selecting the suitable blocks assuring the best robustness and imperceptibility. The blocks are selected after performing a treatment algorithm. These blocks are the key of our algorithm. The steps of this pre-treatment are explained in figure 2.

The preprocessing consists in computing the step 1 to 6 on both the original and compressed-decompressed image. Then, we subtract the middle band of compressed image from the middle band of the original and we select the blocks corresponding to the minimum values. These blocks are the key of our watermarking scheme.

Step9: Inserting the watermark in the LSB of the middle band frequency of the chosen coefficients of the calculated blocks.

In our approach, LSB of these coefficients are substituted by the bits of the watermark after applying Arnold transform.

Step10: Running zigzag inverse, then IDCT to each block to obtain the block data containing the watermark information in the spatial domain.

Step11: Repeating steps 5 to 12 to 'Cb' component. In

ISBN 978-80-86943-41-1

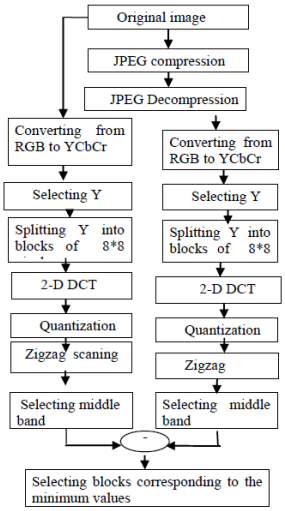


Figure 2: The flow chart of the preprocessing step

addition to Y, Cb is selected to resist to the cropping attack.

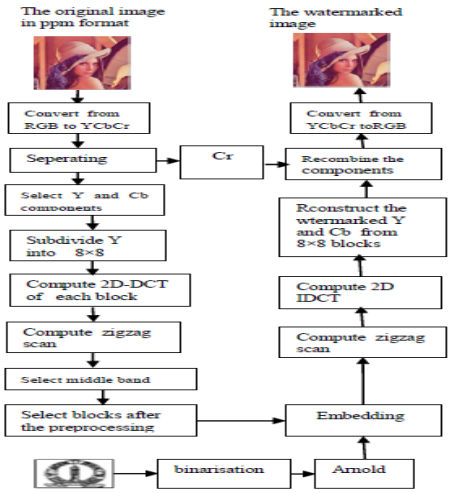


Figure 3: The flow chart of the embedding process

3.2 DCT-LSB-Arnold watermark extracting

The proposed watermarking scheme is a blind approach since the watermark detection doesn't need the original image. The extraction process is depicted in figure 4 and described as below:

Step1: Loading the watermarked RGB image.
Step2: Converting the RGB image to YCbCr color space and separating the Y, Cb and Cr components.
Step3: Choosing the 'Y' component.
Step4: Splitting Y into blocks of 8*8 pixels.
Step5: Each block pixel is subtracted by 128 then transformed into frequency domain using DCT, then quantized and finally scanned by applying zigzag to all 64 DCT coefficients.
Step6: Selecting the middle band frequency coefficients

Step7: Performing the extraction operation using the same key as in the insertion step which is the chosen blocks after pre-treatment.
Step8: Combining the extracted bits together.
Step9: Running the Arnold inverse to obtain the watermark.
Step10: Repeating steps 5 to 9 to 'Cb' component.

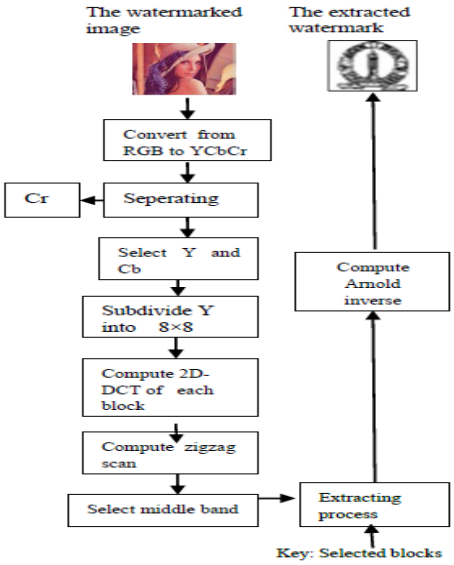
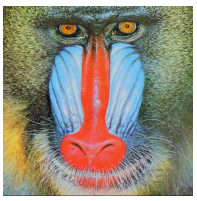


Figure 4: The flow chart of the extraction process

4 EXPERIMENTAL RESULTS

For test evaluation purpose, the experiments are tested on 512x512 standard color images and on 816*1261 Quranic images,available respectively in [LIG17] and [Rea17]. We present some of these images in figure 5.


(a) Lena


(b) Baboon


(c) Peppers


(d) 529


(e) 533

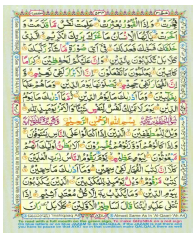

(f) 534

Figure 5: The original images

Figure 6 shows the original image watermark and the results after binarisation step and applying Arnold

ISBN 978-80-86943-41-1

(a) The original watermark

(b) The watermark after binarisation

(c) The watermark after Arnold Transform

Figure 6: The watermark processing

transform 8 times.

The key of our scheme is composed of the number of iteration of the Arnold transform and the index of the chosen blocks in the insertion step.

4.1 Evaluation metrics

The performance of the proposed watermarking scheme is examined and analyzed with three metrics. These metrics are used with reference to the most important requirements of the digital image watermarking : imperceptibility and robustness. The first metric is peak Signal to Noise Rate (PSNR) which is used to measure the quality of the watermarked image with regard to the original one [Nam17]. PSNR in decibels (dB) is given by the following equations:

$$PSNR = 10 \times \log_{10}(L \times \frac{L}{MSE}) \quad (7)$$

In which L is the peak signal amount in the host image and MSE is the mean square error. To distinguish host and watermarked image is usually hard for the human eye. In general, acceptable PSNR values > 36dB [Wke15].

According to (8), the normalized correlation (NC) is used to check the similarity between the original and the extracted watermark bits [Nam17]:

$$NC = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} W(i,j)W'(i,j)}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W(i,j)^2}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} W'(i,j)^2}} \quad (8)$$

In which W (i,j) and W'(i,j) are original and extracted watermark respectively.The original and extracted watermark are similar when NC> 0.75 [Wke15].

According to (9), the bit error ratio (BER) is the ratio showing how many bits are received in error over the number of the total bits received. It is calculated by comparing bit values of both the embedded and the extracted watermark [Nam17].

$$BER = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{W(i,j) \oplus W'(i,j)}{m*n} \quad (9)$$

In which W and W' are original and extracted watermark image respectively, with size of n* m and $\oplus$ means xor operation.

4.2 Imperceptibility results

To test the imperceptibility of our approach we calculate the PSNR. The results of the proposed watermarking scheme for different images are listed in Figure 5. It can be seen from the PSNR values which are over 36 db that the watermarked images have a good quality.

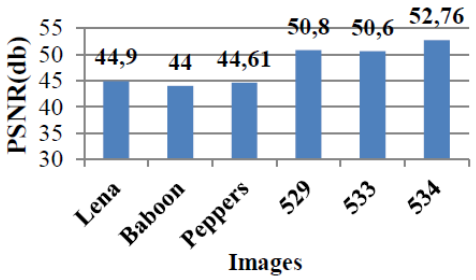


Figure 7: PSNR values of several watermarked images

4.3 Robustness results

A robust watermarking algorithm should resist to different signal distortions and attacks.

To evaluate the performance of the proposed watermark detector against signal manipulations and degradations, we used Stirmark bench 3.1 [Mku] and we calculate the NC and BER values after detection step.

4.3.1 Robustness against JPEG compression

Table 1 shows experimental results from JPEG attacks of Stirmark to the watermarked images 'Lena', 'Baboon', 'Peppers' and to the Quranic images '529', '533', '534'. These images are in '.ppm' format before applying JPEG compression. As shown, our method resists to JPEG attack for rates from 15 to 90. In fact, the NC values are between 0.98 and 1 and the BER values are between 0.01 and 0.

We detect the watermark directly from the compressed images without decompressing them contrary to other techniques which decompress the images before detection [Sra17]. This fact is an important advantage that highlights the originality of our approach.

4.3.2 Robustness against unique attacks of Stirmark

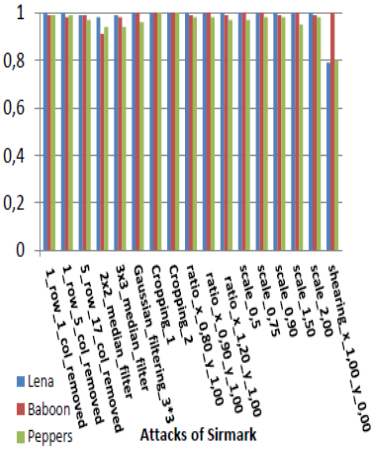
The watermarked images are in ppm formats, the Stirmark generates attacked images in ppm and jpeg formats. The ones with ppm are the unique attacks of Stirmark.

Figure 8 exhibits the values of the NC of the attacked images obtained after applying geometric and combined distortions, the attacked images are 'Lena.ppm', 'Baboon.ppm' and 'Peppers.ppm' and the Quranic imageS which are '529.ppm','533.ppm','534.ppm'.

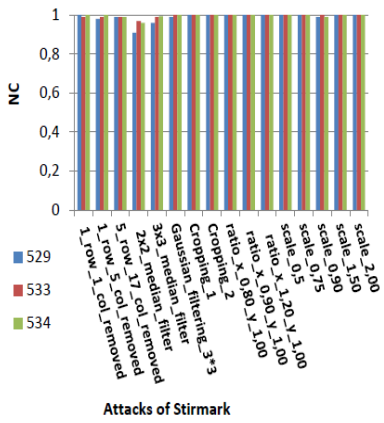
ISBN 978-80-86943-41-1

Image / Quality Factor	Lena		Baboon		Peppers		529		533		534	
	NC	BER	NC	BER	NC	BER	NC	BER	NC	BER	NC	BER
15	1	0	1	0	0.99	0.0009	1	0	0.99	0.001	0.99	0.001
20	1	0	1	0	1	0	1	0	1	0	0.99	0.001
25	1	0	1	0	1	0	1	0	0.98	0.005	0.98	0.006
30	1	0	1	0	0.98	0.006	1	0	0.96	0.013	0.96	0.013
35	1	0	1	0	0.98	0.006	1	0	0.95	0.019	0.99	0.001
50	1	0	0.99	0.003	0.98	0.006	1	0	0.99	0.003	0.98	0.006
70	1	0	1	0	0.99	0.0009	1	0	0.99	0.001	0.99	0.001
80	1	0	1	0	1	0	1	0	1	0	1	0
90	1	0	1	0	1	0	1	0	1	0	1	0

Table 1: Evaluation of the robustness against JPEG compression



(a) The NC values of Lena, Baboon,Peppers images against unique attacks of Stirmark



(b) The NC values of 529, 533,534 images against unique attacks of Stirmark

Figure 8: The NC values against unique attacks of Stirmark

Analyzing the results in Figure 8 we show that our method is robust against several geometric distortions including scaling from 0.5 to 2, symmetric and asymmetric line and column removal, shearing, cropping attack with 1% and 2% and common signal processing including median filter and gaussian filter. In all cases we have obtained NC values greater than the predefined threshold value TNC = 0.75 [Man15] and we detect the watermark without managing to ameliorate the attacked images unlike with other methods [Sra17].
Table 2 confirms the results shown Figure 8, all the values of BER are less than the predefined threshold value TBER = 0.2 [Wke15].

4.3.3 *Robustness against double attacks of Stirmark*

In addition to unique attacks, we evaluate our algorithm against double attacks.
Stirmark Bench combines the distortions with JPEG compression which is known that is very destructive. This brand of combination called double attacks . In table 3, we present the results of our approach against scaling from 0.5 to 2, symmetric and asymmetric line and column removal,common signal processing including median filter, gaussian filter and Frequency mode Laplacian removal (FMLR).

Our proposed scheme successfully resists to double attacks, the highest value of BER values is equal to TBER = 0.2 and we extract the watermark from the attacked image in JPEG format without decompressing it as other methods [Sra17].

4.4 Comparison with existing Algorithms

In this section we present comparison between our method and [Man15] and [Job17] methods.Analyzing the results in Table 4, we show that our algorithm and the work of [Man15] present good robustness against several common signal processing operations, including scaling, aspect ratio and JPEG compression with several quality factors ranging such as 20 and 50. In addition, our proposed method obtains BER values better than [Man15].For scaling attack, we obtain BER equal to 0.001 and 0.009 respectively for scale_0.5 and

ISBN 978-80-86943-41-1

Attacks \ Image name(.ppm)	Lena	Baboon	Peppers	529	533	534
1_row_1_col_removed	0	0	0.0010	0	0.0009	0
1_row_5_col_removed	0	0	0.003	0.008	0.0009	0
5_row_17_col_removed	0.001	0.001	0.008	0.004	0.001	0.0009
2x2_median_filter	0.004	0.03	0.022	0.03	0.03	0.016
3x3_median_filter	0.001	0.008	0.024	0.01	0.0009	0.001
Gaussian_filtering_3*3	0	0	0.012	0.0009	0	0
Cropping_1	0	0	0	0	0	0
Cropping_2	0	0	0	0	0	0
ratio_x_0.80_y_1.00	0	0.0009	0.008	0	0	0
ratio_x_0.90_y_1.00	0	0	0.004	0	0	0
ratio_x_1.20_y_1.00	0	0.0009	0.008	0	0	0
scale_0.75	0	0	0.004	0	0.001	0
scale_0.90	0	0	0.003	0.0009	0	0.009
scale_2.00	0	0	0.002	0	0	0
shearing_x_1.00_y_0.00	0.07	0.07	0.07	0.08	0.08	0.08

Table 2: Evaluation of the robustness against unique attacks of Stirmark

Attacks \ Image name(.jpg)	Lena	Baboon	Peppers	529	533	534
1_row_1_col_removed	0.02	0.03	0.02	0.054	0.07	0.054
1_row_5_col_removed	0.038	0.06	0.04	0.084	0.08	0.083
5_row_1_col_removed	0.041	0.05	0.03	0.074	0.09	0.074
3x3_median_filter	0.05	0.06	0.04	0.057	0.1	0.057
Gaussian_filtering_3*3	0.07	0.078	0.1	0.076	0.2	0.076
FMLR	0.01	0.006	0.03	0.016	0.1	0.016
ratio_x_0.80_y_1.00	0.0068	0.05	0.01	0.075	0.02	0.075
ratio_x_0.90_y_1.00	0.0087	0.02	0.004	0.07	0.03	0.07
ratio_x_1.20_y_1.00	0.006	0.019	0.02	0.025	0.02	0.025
scale_1.10	0.044	0	0.09	0.059	0.1	0.059
scale_1.50	0.003	0	0.01	0.003	0.03	0.003
scale_2.00	0	0	0.006	0	0.01	0

Table 3: Evaluation of the robustness against double attacks of Stirmark

Attacks \ Methods	Proposed method **PSNR=49**		[Man15] **PSNR=44**	[Job17] **PSNR=38**	
	NC	BER	BER	NC	BER
No attack	1	0	0	1	0
Scale_0.5	0.99	0.001	0.02	0.86	0.16
Scale_2	0.99	0.009	0.02	0.86	0.16
Aspect ratio(1.2, 1.0)	0.99	0.001	0	-	-
JPEG (Q=20)	0.99	0.001	0.003	0.73	0.34
JPEG (Q=50)	0.99	0.001	0.001	0.83	0.22
JPEG (Q=60)	0.99	0.001	-	0.75	0.32

Table 4: Comparison of robustness between the proposed approach and other existing image watermarking schemes

scale_2, [Man15] obtain BER =0.02 for both attacks.
On the other hand, the method proposed in [Job17] has a similar performance against the above mentioned common signal processing. However, the method seem to be not robust to JPEG compression, obtaining a BER =0.22, and 0.34.
The two methods don't deal with double attack problem like ours.

 ISBN 978-80-86943-41-1

5 CONCLUSION

In this paper, we focused on image watermarking for copyright protection application. A DCT-based blind and robust image watermarking scheme using the least significant bit is presented. In order to increase the security, the Arnold transform is used to scramble the watermark. In the proposed scheme the Y and Cb components of the original image are substituted by the bits of the scrambled watermark.
The results demonstrate high imperceptibility and robustness against JPEG compresion , unique and double attacks of Stirmark without ameliorating the watermarked attacked image before detection. At the end, our method is compared to other existing algorithms, the method presents good results.

6 ACKNOWLEDGMENTS

The research leading to these results received funding from the Tunisian Ministry of Higher Education and Scientific Research under the grant agreement number LR11ES48.

7 REFERENCES

[Aro15] A.Roy, A. K. Maiti, and K. Ghosh, A perception based color image adaptive watermarking scheme in YCbCr space, in Signal Processing and Integrated Networks, 2nd International Conference, 2015.

[Ali10] Ali Wali, Najib Ben Aoun, Hichem Karray, Chokri Ben Amar, Adel M. Alimi: A New System for Event Detection from Video Surveillance Sequences. Advanced Concepts for Intelligent Vision Systems International Conference, ACIVS, 2010.

[Bou11] Boulbaba Guedri, Mourad Zaied, Chokri Ben Amar: Indexing and images retrieval by content. High Performance Computing and Simulation (HPCS), 2011.

[Fat15] Faten Chaabane, Maha Charfeddine,William Puech,Chokri Ben Amar, A qr-code based audio watermarking technique for tracing traitors, EUSIPCO, 2015.

[Hai13] Hai Tao, Li Chongmin, Jasni Mohamad Zain1, Ahmed N. Abdalla,Robust Image Watermarking Theories and Techniques: A Review,Journal of Applied Research and Technolog, 2014

[Job17] Jobin Abraham ,Varghese Paul, An imperceptible spatial domain color image watermarking scheme, Journal of King Saud University – Computer and Information Sciences, 2017.

[Lam15] Lamri Laouamer and Omar Tayan, A Semi-Blind Robust DCT Watermarking Approach for Sensitive Text Images, Arab J Sci Eng, 2015.

[LIG17] LIGM, Laboratoire d'Informatique Gaspard-Monge,1990 http://igm.univmlv.fr/ incerti/IMAGES/PPM.htm. Online accessed April 2017.

[Mah12] Maha Charfeddine, Maher El'arbi and Chokri Ben Amar, A new DCT audio watermarking scheme based on preliminary MP3 study , Multimed Tools Appl, 2012.

[Man16] Manpreet Kaur, Vinod Kumar Sharma, Encryption based LSB Steganography Technique for Digital Images and Text Data, International Journal of Computer Science and Network Security, 2016

[Man15] Manuel Cedillo-Hernandez , Antonio Cedillo- Hernandez Francisco Garcia-Ugalde, Mariko Nakano-Miyatake, Hector Perez-Meana, Copyright Protection of Color Imaging Using Robust-Encoded Watermarking, Radioengineering, 2015.

[Mas10] Masmoudi Salma, Charfeddine Maha, Chokri Ben Amar A robust audio watermarking technique based on the perceptual evaluation of audio quality algorithm in the multiresolution domain,Signal Processing and Information Technology (ISSPIT), 2010

[Meh16] Mehdi Khalili and Mahsa Nazari, Non Correlation DWT Based Watermarking Behavior in Different Color Spaces , International Journal of Advanced Computer Science and Applications, 2016.

[Mel11] M.El'Arbi, M.Charfeddine,S. Masmoudi M.Koubaa and C. Ben Amar,Video watermarking algorithm with BCH error correcting codes hidden in audio channel,IEEE symposium series in computational intelligence, 2011

[Mku] M.Kutter and F. Petitcolas, A fair benchmark for image watermarking systems,SPIE, 1999

[Moh16] Mohammad Moosazadeh and Gholamhossein Ekbatanifard, Robust Image Watermarking Algorithm Using DCT Coefficients Relation in YCoCg-R Color Space ,Eighth International Conference on Information and Knowledge Technology (IKT), Hamedan, Iran,2016.

[Nam17] Namita Tiwari1 and Sharmilaand, Digital Watermarking Applications, Parameter Measures and Techniques, International Journal of Computer Science and Network Security, 2017.

[Pooj15] Pooja Rani , Apoorva Arora, Image Security System using Encryption and Steganography, International Journal of Innovative Research in Science, Engineering and Technology,2015.

[Rea17] Reading Al Quran, 2017, http://readingalquran.com/alquran.php?part=30, Accessed April 2017

 ISBN 978-80-86943-41-1

[Sra17] S. Rashmi, Priyanka and Sushila Maheshkar,Robust Multiple Composite Watermarking Using LSB Technique , Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Advances in Intelligent Systems and Computing, 2017

[Ssh14] S.Shanmugaprabha and N.Malmurugan, A New Robust Image Watermarking Scheme Based On DWT With SVD , International Journal of advanced studie in Computer Science and Engineering, 2014.

[Son16] Sonam Tyagi, Harsh Vikram Singh, Raghav Agarwal and Sandeep Kumar Gangwar,Digital Watermarking Techniques for Security Applications,International Conference on Emerging Trends in Electrical, Electronics and Sustainable Energy Systems, 2016.

[Sub17] Subin Bajracharya and Roshan Koju, An Improved DWT-SVD Based Robust Digital Image Watermarking for Color Image International Journal Engineering and Manufacturing, pp 49-59, 2017

[Sur17] Suraj Kumar Dubey, Vivek Chandra, Steganography, Cryptography and Watermarking: A Review , International Journal of Innovative Research in Science, Engineering and Technology, 2017.

[Wke15] W.K. ElSaid, Watermarking Digital Artworks , International Journal of Computer Applications,2015.

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

Optimizing Spectral Fresnel Reflectance for Displays

Jonathan Brian Metzgar
University of Alaska Fairbanks
College of Engineering and Mines
PO Box 755960
Fairbanks, Alaska 99775 USA
jmetzgar@acm.org

Sudhanshu Kumar Semwal
University of Colorado at Colorado Springs
Department of Computer Science
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918 USA
ssemwal@uccs.edu

ABSTRACT

Approximate equations for rendering Fresnel reflectance abound in computer graphics. We take a fresh approach and consider not only the approximation but the display device as well. The sRGB color standard is finally giving way to wide color spaces such as Adobe RGB and DCI P3 which display more color. We present a preprocessing method to use measured spectral index of refraction data and the color space specification to synthesize an RGB complex index of refraction. Metals, in particular, generally require a spectral renderer, but we created a way to sample the complex index of refraction and absorptive index that acts like the color filter built into the display. Our novel contribution uses a normal distribution centered around the ideal display red, green, and blue wavelengths derived from the CIE xy coordinates and respective white point to window sample the complex index of refraction. We created a WebGL experimental platform that uses the Schlick inspired Lazanyi and Szirmay-Kalos (LSK) multispectral Fresnel approximation coupled with modern physically based BRDFs to simulate the appearance of metal. Our application can compare five different Fresnel implementations coupled with physically based Blinn-Phong and GGX microfacet models. We demonstrate that with reasonable filter widths, we eliminate the need for a spectral renderer for real-time rendering. Additionally, we utilize publicly available measurement data to simulate a variety of metals ranging from silver, gold, and copper to silicon, lead, and zinc.

Keywords

Fresnel reflectance modeling, rasterization, physically based rendering, complex index of refraction, absorptive index

1 INTRODUCTION

Fresnel reflectance is a critical component in many microfacet based bidirectional distribution reflectance functions (BRDF). It provides the effect that viewing angles close to the normal provide the highest translucency into a refractive surface or at grazing angles, the highest reflectance of the environment. For conductors or metals, it provides their natural color and reflective properties. However, the equations are computationally expensive, so a variety of approximations exist for high-performance applications like real-time rendering.

The Fresnel equation is a function using the index of refraction, itself dependent on wavelength. For many materials, the index of refraction is consistent for the visible electromagnetic spectrum. For some metals such as gold and copper, there is significant absorption in the green, blue, and violet bands. Merely using a constant factor for the entire visible spectrum will introduce errors. However, using separate refractive indices for red, green, and blue may introduce color errors when not targeted for the color space of the intended display device. The state-of-the-art is to use spectral rendering,

impractical for many real-time applications, to accurately reproduce these types of materials.

Because spectral rendering is difficult for real-time applications, we propose synthesizing the complex index of refraction using a chosen color space. Our novel contribution is a preprocessing step that uses window sampling using a Gaussian filter of the refractive index data for the red, green, and blue (RGB) wavelengths. We use the CIE Spectral Locus (or "horseshoe") and a line-distance function to determine the proper RGB wavelengths. Finally, we implemented an HTML5 application using TypeScript and WebGL to test a variety of exact and approximate Fresnel equations and enable direct comparison for the accuracy of each method. Our results show that using our simple method enables developers to create real-time renderings of metal without needing a spectral renderer.

2 PREVIOUS WORK

Several background ideas must be considered for the realistic rendering of metals and other wavelength dependent materials. There is the Fresnel equation and several approximations which we review in this section. There is also the reflection model or microfacet BRDFs which we cover separately in the next section. Color theory and perception are essential because our method relies on using several color space specifications for determining the appropriate RGB wavelengths. We introduce mathematical equations like the Gaussian filter and relative distance function when they are needed in this paper.

Schlick's approximation [Sch94],

$$F(\theta_d) = F_0 + (1 - F_0)(1 - \cos^5 \theta_d),$$

or,

$$F(\eta_1, \eta_2, \theta_d) = \frac{(1-\eta_2)^2 + 4\cos^5 \theta_d \eta_2}{(1+\eta_2)^2}$$

where F_0 is the base reflectance at incidence 0° and commonly used in rendering algorithms because it has low computational cost. Recently, this work was been extended to support metals [LS05] by including the absorptive index κ,

$$F(\eta_1, \eta_2, \theta_d) = \frac{(1-\eta_2)^2 + 4\cos^5 \theta_d \eta_2 + \kappa_2^2}{(1+\eta_2)^2 + \kappa_2^2}.$$

It is also common to precalculate F_0 for red, green, and blue in Schlick's approximation [AMHH08] using the full equation and eliminates the reddish hue that was identified by Lazányi and Szirmay-Kalos [LS05]. For briefness, we refer to their method as the LSK method.

A common approximation used by Glassner [Gla95] and others [PH10] for the Fresnel equations is

$$\rho_\parallel^2 = \frac{(\eta_2^2 + \kappa_2^2)\cos^2\theta_d - 2\eta_2 \cos\theta_d + 1}{(\eta_2^2 + \kappa_2^2)\cos^2\theta_d + 2\eta_2 \cos\theta_d + 1}, \quad (1)$$

$$\rho_\perp^2 = \frac{(\eta_2^2 + \kappa_2^2) - 2\eta_2 \cos\theta_d + \cos^2\theta_d}{(\eta_2^2 + \kappa_2^2) + 2\eta_2 \cos\theta_d + \cos^2\theta_d}, \quad (2)$$

$$F = \frac{\rho_\parallel^2 + \rho_\perp^2}{2}. \quad (3)$$

The full form of the Fresnel equations for an absorbing medium [Mod13, p.51-53] can be written as

$$\rho_\parallel = \frac{(p - \sin\theta_1 \tan\theta_1)^2 + q^2}{(p + \sin\theta_1 \tan\theta_1)^2 + q^2}\rho_\perp,$$

$$\rho_\perp = \frac{(\cos\theta_1 - p)^2 + q^2}{(\cos\theta_1 + p)^2 + q^2}$$

where

$$p^2 = \tfrac{1}{2}\sqrt{\left(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2\theta_1\right)^2 + 4\eta_2^2\kappa_2^2} + \tfrac{1}{2}\left(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2\theta_1\right),$$

$$q^2 = \tfrac{1}{2}\sqrt{\left(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2\theta_1\right)^2 + 4\eta_2^2\kappa_2^2} - \tfrac{1}{2}\left(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2\theta_1\right).$$

If we let

$$a = \eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2\theta_1,$$

then we can shorten the definition for p^2 and q^2 to

$$p^2 = \frac{1}{2}\left[\sqrt{a^2 + 4\eta_2^2\kappa_2^2} + a\right],$$

$$q^2 = \frac{1}{2}\left[\sqrt{a^2 + 4\eta_2^2\kappa_2^2} - a\right].$$

Lastly, the angle of refraction is given by the *Generalized Snell's Law* equation

$$p \tan\theta_2 = \eta_1 \sin\theta_1.$$

Color space theory [JG78, MG80] provides insight into the perception of color and how displays are designed to display color. For example HP and Microsoft introduced the sRGB profile [IEC99] which gives guidance on how to design a display device such that a certain range of colors can be accurately produced. Although sRGB achieved widespread use, it has a limited range of color. Wider color spaces such as Adobe RGB [Ado98] and DCI-P3 [SMP11] are popular amongst artists and for modern consumer devices.

These color spaces use CIE 1931 xy coordinates to define the domain of colors (shown in Figure 1). The white point determines the wavelength of light for each of the three primaries. The wavelengths of light are

ISBN 978-80-86943-41-1

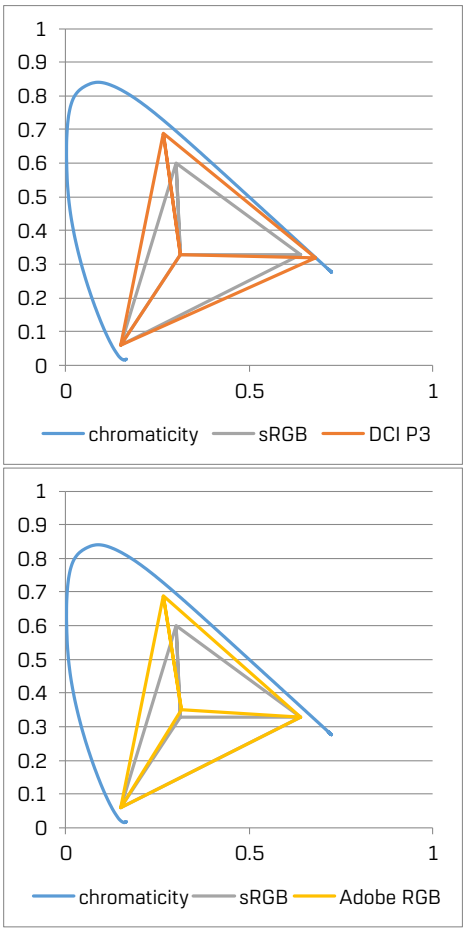


Figure 1: A comparison of sRGB (gray), DCI-P3 (orange), and Adobe RGB (yellow) color spaces.

dependent on the white point and the red, green, and blue coordinates. The line intersecting the spectral locus (the outer curved line representing wavelength), the chromaticity coordinate xyY, and the white point determines the wavelength because we are moving from the white point towards monochromaticity. This is called the *dominant wavelength* or *equi-hue line*. Later, we show that we can apply a relative distance function to find the wavelength if we only have the raw CIE horse shoe data and color space chromaticity coordinates. Figure 2 compares the sRGB and DCI P3 color spaces and the sRGB and Adobe RGB color spaces.

3 APPLYING FRESNEL REFLECTION

The rendering equation [Kaj86] by Kajiya,

$$\mathbf{L}_o(\mathbf{x} \to \omega_o) = \mathbf{L}_e(\omega_o \to \mathbf{x}) + \int_\Omega f_r(\omega_i, \omega_o)\, \mathbf{L}_i(\omega_i \to \mathbf{x})\, \langle \omega_i, \omega_g \rangle \, \mathrm{d}\,\omega_i \,,$$

says that the light approaching the viewer from point $\mathbf{x}$ is proportional to all the light incident from the positive hemisphere. In other words, every incoming direction at $\mathbf{x}$ is a potential source of light. In turn, the amount of reflected light in the direction ω_o is determined by the bidirectional reflectance distribution function (BRDF) [Nic65]

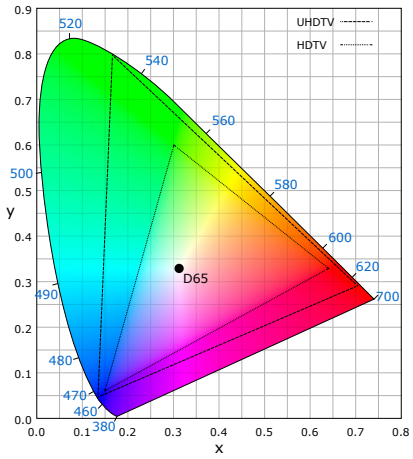


Figure 2: The CIE 1931 xy color space horse shoe [Com16] comparing the Rec 2020 and Rec 709 standards for UHDTV and HDTV.

$$f_r(\omega_i, \omega_o) = \frac{\mathrm{d}\,L_o(\omega_o)}{\mathrm{d}\,E_i(\omega_i)} = \frac{\mathrm{d}\,L_o(\omega_o)}{L_i(\omega_i)\cos\theta_i\,\mathrm{d}\,\omega_i} \,.$$

For glossy surfaces using microfacet models, the widely used Cook-Torrance model [CT81] is

$$f_r(\omega_i, \omega_o) = \frac{D(\omega_h)\, F(\theta_d)\, G_2(\omega_i, \omega_o)}{4\cos\theta_i \cos\theta_o} \,,$$

where $D(\omega_h)$ is the microfacet distribution, $F(\theta_d)$ is the Fresnel factor, and $G_2(\omega_i, \omega_o)$ is the masking-shadowing function. The five critical vectors necessary to evaluate this function is the incoming vector ω_i, outgoing vector ω_o, geometric normal ω_g, half angle vector $\omega_h = \omega_i + \omega_o$, and difference angle vector $\omega_d = \omega_i + \omega_h$. We use the *difference angle* $\theta_d = \arccos \omega_i \cdot \omega_h$ is used for Fresnel. To be physically based, it must obey the laws of conservation of energy, Helmholtz reciprocity, and positivity. A thorough analysis of the BRDF and masking-shadowing function is shown by Heitz [Hei14]. Diffuse reflection models, such as the Disney BRDF, also incorporate a Fresnel term [BS12]. This is especially helpful for rendering soft materials like felt or velvet.

We used both the Blinn-Phong and generalized GGX microfacet distributions for testing purposes. The roughness of the surface is given by α. The normalized Blinn-Phong microfacet model is

$$D(\omega_g, \omega_h) = \frac{1}{\pi\alpha^2}(\omega_g \cdot \omega_h)^{\frac{2}{\alpha^2+\varepsilon}-(2+\varepsilon)}$$

where ε is a small offset to eliminate division by zero when the roughness is zero. The GGX microfacet distribution [BS12, WMLT07] is

$$D_{\mathrm{GTR}}(\omega_g, \omega_h) = \frac{1}{\pi}\left(\frac{1}{(\alpha^2-1)(\omega_g \cdot \omega_h)^2 + 1}\right)^\gamma$$

ISBN 978-80-86943-41-1

Figure 3: Monochromatic choice of Fresnel sampling can result in deviations. Compare blue wavelengths of 445nm (left) and 500nm (right)

Figure 4: A comparison of gold when rendered with sRGB (left) and Adobe RGB (right) showing that color space is important for rendering.

where γ is an adjustable parameter that closely matches the Beckmann distribution at 10 and the Trowbridge-Reitz at 2.

The masking-shadowing functions use the G_2 version (see Heitz [Hei14] for more information) which take into account both incoming and outgoing directions. The Smith function

$$G_2(\omega_i, \omega_o, \omega_g) = \frac{1}{1+\Lambda(\omega_i)+\Lambda(\omega_o)}$$

and the GGX Smith function is

$$\Lambda(\omega) = \frac{-1+\sqrt{1+\frac{(\omega_g\cdot\omega)^2}{\alpha^2(1-(\omega_g\cdot\omega)^2)}}}{2}.$$

The Blinn-Phong masking-shadowing function [Bli77] is

$$G_2(\omega_i, \omega_o, \omega_g) = \min\left\{1, \frac{(\omega_g\cdot\omega_h)(\omega_g\cdot\omega_o)}{\omega_o\cdot\omega_h}, \frac{(\omega_g\cdot\omega_h)(\omega_g\cdot\omega_i)}{\omega_o\cdot\omega_h}\right\}.$$

4 THE APPEARANCE OF METAL

The appearance of metal is directly related to the effect of the Fresnel equation. If we have a spectral measurement of the index of refraction η and its absorptive index κ, we can reconstruct its color. So the problem we wish to solve is to choose which wavelength of light to use for display. It might seem implausible, at least initially, to synthesize the index of refraction, but that is exactly what we must do. And therefore we must understand how the display works.

Any modern LCD display is a composition of white light, liquid crystals, and color filters. The liquid crystals control how much light gets to the color filters. We understand that the backlight is not perfectly pure, but it must be good enough for accurate reproduction of

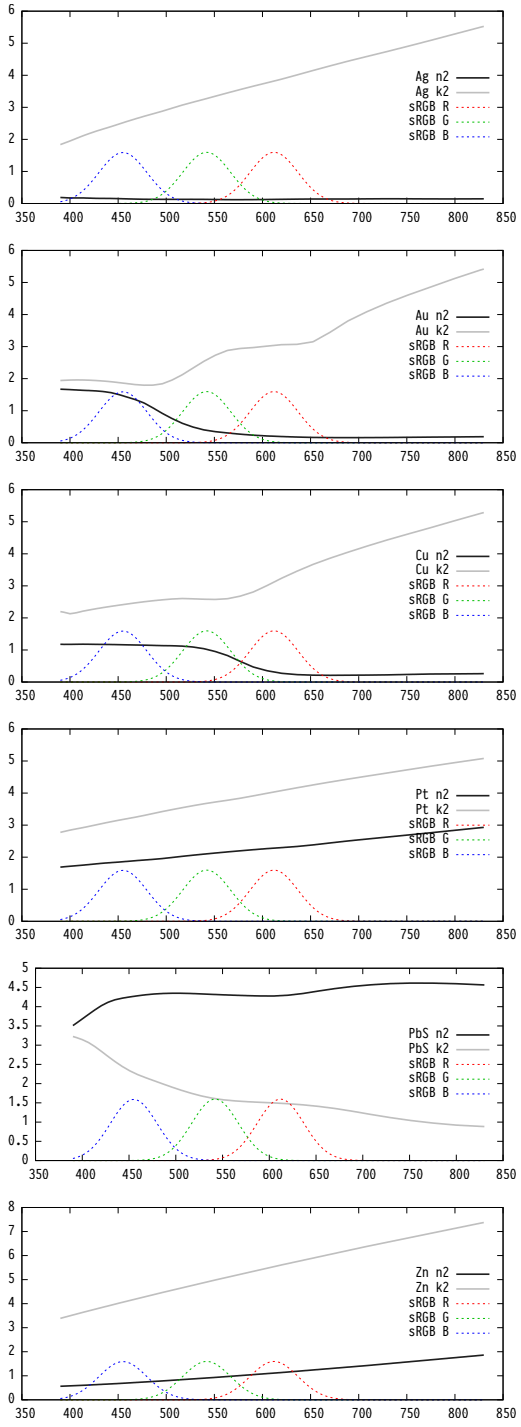


Figure 5: Comparison of index of refraction for materials Ag, Au, Cu, Pt, PbS, and Zn. Dark grey is used for η and light gray for κ. Wavelengths go from 390nm to 830nm.

ISBN 978-80-86943-41-1

Figure 6: Comparison of Fresnel using irradiance only. Clockwise from top left: Silver, Gold, Platinum, Zinc, Lead, and Copper.

Figure 7: Comparison of Fresnel with reflections added. Clockwise from top left: Silver, Gold, Platinum, Zinc, Lead, and Copper.

color. The backlight and filters may affect the quality of color reproduction and many professionals choose calibrated monitors for this reason. Calibrated monitors usually have a lookup table so they can target several color spaces. Each designer of any display will choose a light source, liquid crystal display, and color filters depending on factors such as intended audience and target price. Our insight is that the color filters act like Gaussian filters for a given wavelength, rather than just band pass a single wavelength.

So the quality of the display is a function of how good the white light is and the effectiveness of the color filters. The best display systems are calibrated and measured against standards of gamma correction, and matched against the color space specification such as sRGB, DCI P3, and Adobe RGB. Figure 4 shows that each color space needs to be handled specially if there is a significant change in white point or red, green, and blue *xy* coordinates.

Let us consider the effect of the color filters on the index of refraction. Consider that instead of the white light and the LCD that we only have a source of light, the metal, and the color filter. The color filter will attenuate the reflection of light from the metal according to its filter shape. A red filter may be a band-pass or low pass filter, the green filter a band-pass filter, and the blue filter either a band-pass or high-pass filter. Let us consider only band-pass filtering as our eyes effectively constrain the lower and upper wavelengths of visible light.

It would be very easy to ignore the color filter design of the display and choose wavelengths based on a set of coordinates. For example, the CIE RGB space defines red as 700nm, green as 546.1nm, and blue as 435.8nm. But sRGB is 612nm, 542nm, and 455nm and Adobe RGB is 612nm, 527nm, and 455nm.

The BRDF in the rendering equation relates how much light is reflected for each of the red, green, and blue wavelengths. The color filter will have the effect of increasingly attenuating how much reflected light is seen as λ moves away from the peak reflection wavelength λ_{filter}. Spectral rendering, which considers the entire visible spectrum, would be the natural choice in ensuring an accurate rendering. To produce a spectral renderer, numerous wavelengths of light are used to evaluate the rendering equation. (As a nod to Newton, we would not be rendering with RGB, we would be rendering with ROYGBIV.) Path tracing is the obvious approach for spectral rendering because it is easy to incorporate into the design. Then once all the samples have been created for the pixel, it is sampled to construct an RGB value. A Gaussian filter is a natural choice to sample the spectral values. More advanced techniques could even pair a tristimulus model to take into account human color perception.

Let us say that λ_c represents the color we want to sample, where c is red, green, or blue. A spectral evaluation for the Fresnel equations for a given wavelength λ_c using the normal Gaussian filter [Bul79] is

$$F_{\lambda_c}(\theta) = \frac{1}{\sigma\sqrt{2\pi}} \int_{\infty}^{-\infty} F(\lambda,\theta) \exp - \frac{(\lambda-\lambda_c)^2}{2\sigma^2} d\lambda$$

where σ is the standard deviation of the Gaussian filter. Our filter width is approximately 6σ which is 99.7% of all the area under the Gaussian curve. According to probability theory, 68.27% of the area is between $\pm\sigma$, 95.45% is between $\pm 2\sigma$, and 99.73% is between $\pm 3\sigma$.

ISBN 978-80-86943-41-1

If we set σ to be small, only wavelengths around the center wavelength have significant contribution to the final result. This can be seen in Figure 5 where σ is set to 25nm.

We now have two paths we can approach. We can pick three wavelengths and evaluate the Fresnel equations or we can use all wavelengths and spectrally evaluate the Fresnel equations. We propose that instead of evaluating the Fresnel equation at all wavelengths or only choosing three wavelength values, that we instead sample the index of refraction around the wavelengths of the intended display color specification.

The reason for proposing this is twofold. The first reason is that spectral rendering is impractical for many realtime applications, such as mobile devices. And second, for some materials, such as gold or copper, the choice of wavelength for red, green, and blue causes significant changes in appearance. Figure 3 shows that changing blue causes the reflection of the sky to change color.

Before we get into the implementation details, let us quickly describe how our method would be applied. First, you would get access to wavelength dependent index of refraction measurements. Second you would resample the data so that it is in even steps (we used steps of $\frac{1}{10}nm$). Third, you would determine the wavelengths of red, green, and blue that are used for the display by using the color specification. Fourth, you would use a Gaussian distribution to sample the index of refraction. Lastly, you could calculate F_0 and use an approximation, like specular color or Schlick's approximation, or you could use a higher precision formula (as we mentioned in section 2). And now let us look at our implementation.

5 IMPLEMENTATION

We implemented a TypeScript and WebGL based renderer in a HTML5 single page application[1] to use all the rendering techniques except for the spectral renderer. We can index an environment map to test both reflections and irradiance. As with most realtime rendering techniques, only a subset of light paths are considered. There is no performance penalty in using our method since it is a preprocessing step for the material. The performance is directly related to the Fresnel equation or approximation chosen.

We use three light paths. First we use the Sun as the primary light source using astronomical calculations to determine a realistic position in the sky. We also calculate the position for the Moon (a simple model without perturbations) for night lighting conditions. This

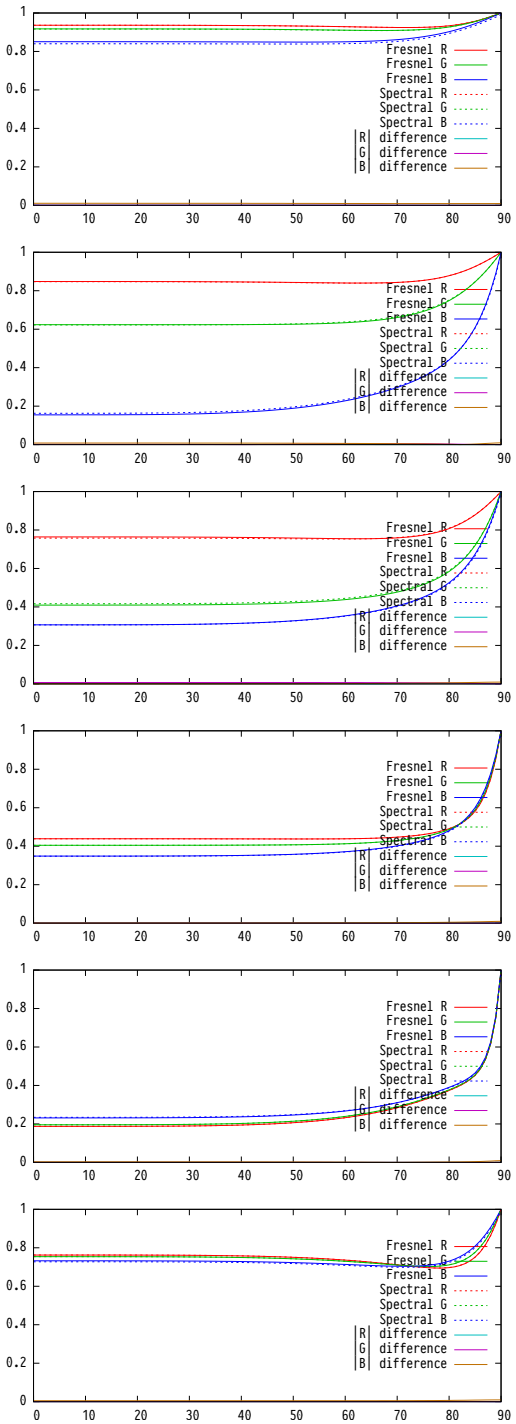


Figure 8: Comparison of relative error of our approximation to the spectral Fresnel functions for materials Ag, Au, Cu, Pt, PbS, and Zn. Wavelengths go from 390nm to 830nm.

[1] We extended our HTML5 web application to generate Gnuplot data and charts that can be copy-pasted to create the figures in this paper.

ISBN 978-80-86943-41-1

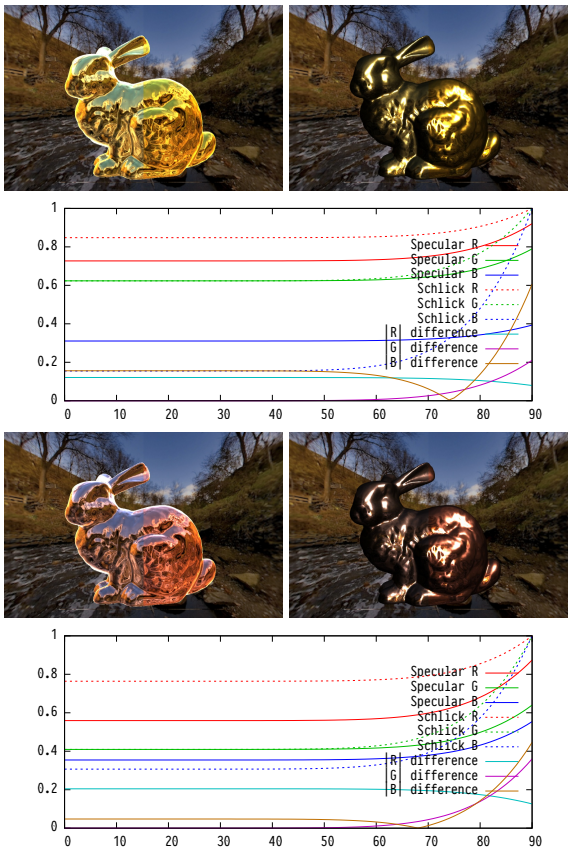


Figure 9: A comparison of using a Fresnel derived specular color k_s (image left halves) and single channel Schlick's approximation (right image halves). The error is shown in the graphs. Gold is top and Copper is bottom with burnished (left) and dull (right) finishes.

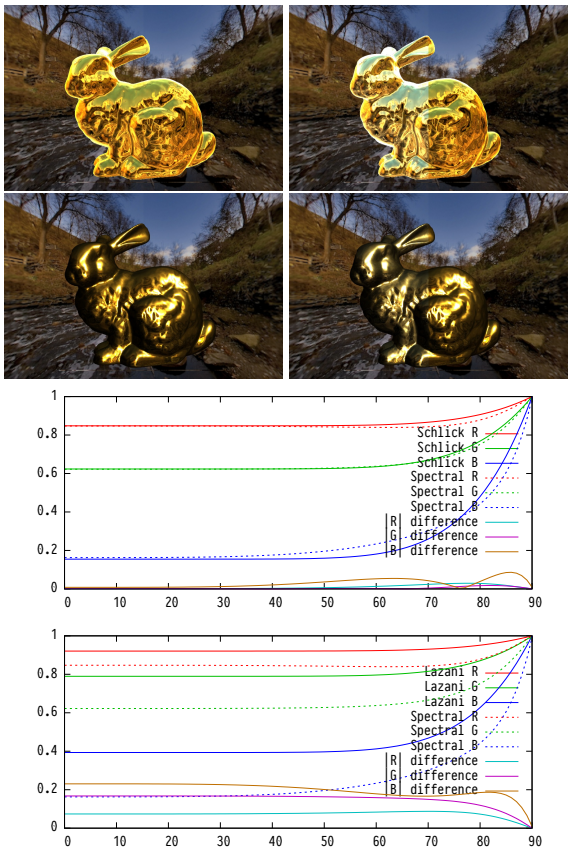


Figure 10: A comparison of Schlick's (left) and LSK method (right) for Gold. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

also allows for us to animate the motion of the sun to observe the change in reflection. Secondly, we use an environment map to simulate reflection and irradiance. We calculate the reflection vector using the view vector and the normal. We calculate irradiance by using the geometric normal ω_g to index the environment map. Figures 6 and 7 show dull (irradiance) and burnished (reflections) imagery using the sRGB color space for a variety of metals. Charts of their index of refraction are shown in Figure 5.

There are publicly available spectral measurements for a variety of materials [fil18]. Our test application has measurements for many common metals including Silver (Ag), Aluminum (Al), Gold (Au), Chromium (cr), Copper (Cu), Nickel (Ni), Lead Sulfide (PbS), Platinum (Pt), Silicon (Si), Titanium (Ti), Tungsten (W), Zinc (Zn), and Schott SF6HT (glass). We picked these because they have complete spectral profiles.

We allow the user to select from a number of color specifications including sRGB, DCI P3, and Adobe RGB. There is a monochromatic option where the user can set the wavelength individually for red (650 to 740nm), green (520 to 565nm), and blue (435 to 500nm) with a custom width for the Gaussian σ standard deviation. At 0, we use a Dirac delta function to only use the specified wavelength, otherwise we use the normal distribution.

Now we use the color space profile to determine the red, green, and blue wavelengths. We can identify which wavelength of light to use by tracing a line from the white point coordinates w_{xy} through the chromaticity coordinates c_{xy}, where c is red, green, or blue, and find the intersection on the CIE horse shoe which is mapped to wavelength. We can use the relative distance formula

$$d = \frac{y(\mathbf{v}_{2,y} - \mathbf{v}_{1,y}) - x(\mathbf{v}_{2,x} - \mathbf{v}_{1,x}) + \mathbf{v}_{2,x}\mathbf{v}_{1,y} - \mathbf{v}_{2,y}\mathbf{v}_{1,x}}{||\mathbf{v}_2 - \mathbf{v}_1||}.$$

to calculate the distance to each of the CIE horse shoe points until we minimize d. This method generalizes finding the wavelength for a given white point and chromaticity coordinate. We believe this is a novel contribution of our method.

Next we prepare to resample the wavelengths. We create an array with indexes 3900 to 8300 to correlate to wavelengths 390nm to 830nm, giving 4,401 data points. The data structure contains the wavelength, index of

ISBN 978-80-86943-41-1

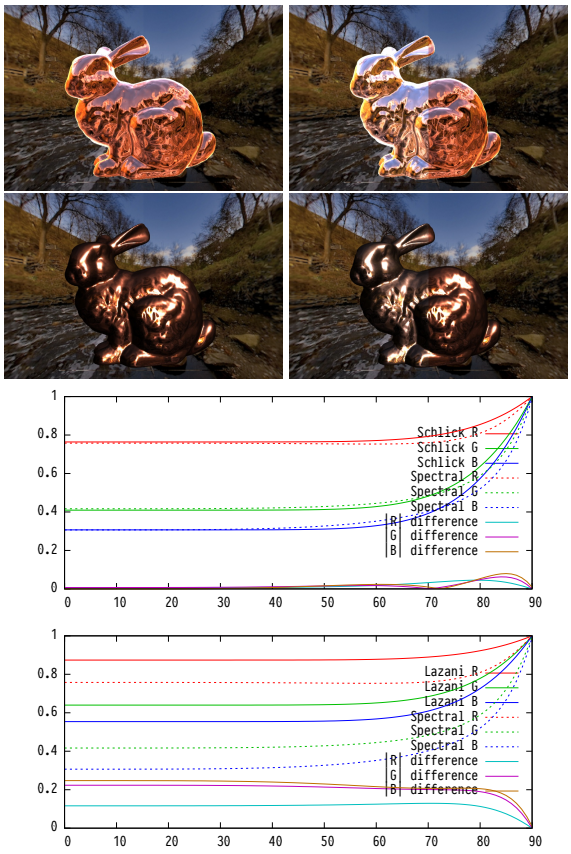


Figure 11: A comparison of Schlick's (left) and LSK method (right) for Copper. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

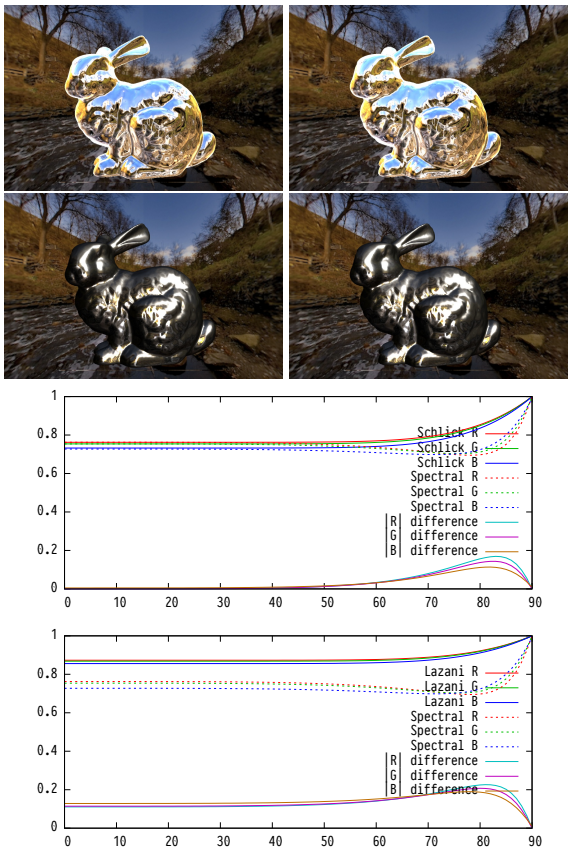


Figure 12: A comparison of Schlick's (left) and LSK method (right) for Zinc. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

refraction, absorptive index, and RGB weight for the normal distribution function. Our purpose for using 10 samples per nanometer of wavelength is to avoid the situation where we use too few samples statistically. For example, If we use a standard deviation of 12nm, that's only 24 samples–so we increase it by a factor of 10 to avoid that. But first we need to preprocess the input index of refraction data.

We use linear interpolation to resample the original measurements so that they are specified for each tenth of a nanometer in an array. We have not analyzed whether using nearest, linear, or higher forms of sampling would make a huge difference. We think linear is fine unless you need continuous derivatives, which we do not. Then we create a normal distribution calculation for each wavelength data point. If $G(\lambda_i, \sigma^2)$ is the normal distribution function, we then resample the index of refraction

$$\eta' = \frac{1}{10} \sum_{i=3900}^{8300} \eta_i G(\lambda_i, \sigma^2)$$

and absorptive index

$$\kappa' = \frac{1}{10} \sum_{i=3900}^{8300} \kappa_i G(\lambda_i, \sigma^2)$$

for red, green, and blue. We divide by 10 in order to account for using 10 samples per each nm of wavelength. Finally, we use these resampled values for inputs in the Fresnel equation in the shader.

6 RESULTS

If the standard deviation for red, green, and blue is kept reasonable (~25nm), then the error using our method compared with the spectral rendering is acceptable. Figure 8 and Figure 13 shows three separate error lines for red, green, and blue. The error line is the absolute error at each angle for the Fresnel function. Our results show that the index of refraction can be re-synthesized with negligible visible error.

Our method also supports comparing the different approximations to the spectral version. We noticed in Lazányi and Szirmay-Kalos (LSK) [LS05] that they used a multispectral approximation, noting that

ISBN 978-80-86943-41-1

Schlick's approximation yielded red tinted results. We understand that this is caused by ignoring the absorptive index which often is significantly greater than the index of refraction. The common way of using Schlick's approximation is to use an accurate Fresnel calculation at 0°, and we show that this is better than the LSK model.

We also note that a common way to approximate gold (i.e. cheating, hacking, or sometimes gross approximations) is to use the specular color k_s. So we present a similar approach and set the specular color $k_s = (F_{0,\mathrm{r}}, F_{0,\mathrm{g}}, F_{0,\mathrm{b}})$ to the base reflection color and then use their approximation using either the average η and κ or even just the green index since our eyes are most sensitive to green. We can use a Dirac-Delta function to eliminate the integral if we are interested in only a single angle (e.g. θ_d) and so the radiance is

$$L_o = k_s L_i(\omega_g \cdot \omega_i) \frac{F(\theta_d) D(\omega_h) G(\omega_i, \omega_o)}{4 \cos\theta_i \cos\theta_o}$$

and

$$F(\theta_d) = F_0 + (1 - F_0)(1 - \cos^5 \theta_d)$$

where

$$F_0 = \frac{F_{\mathrm{r}}(0°) + F_{\mathrm{g}}(0°) + F_{\mathrm{b}}(0°)}{3}$$

or

$$F_0 = F_{\mathrm{g}}(0°).$$

We show in Figure 9 the copper material using the full Fresnel equation and the specular color approximation. This gets rid of the tinting that Schlick's approximation inherently had (if you mistakenly ignored the complex index of refraction), but at the expense of either underestimating or overestimating reflection at grazing angles. It is not very good and we would not recommend using anything less than the LSK spectral approximation. Perhaps our specular color version may be useful for slow GPUs where only a single channel of Fresnel can be computed.

Schlick's approximation, the approximate, and the exact Fresnel equations look almost identical in practice. LSK is sometimes brighter or dimmer and requires an error compensation factor of

$$-ax(1-x)^{\alpha}$$

but α was not fully explained in their paper. The idea is to compensate for materials like Zinc or Aluminum which have a dip in their reflectance at near glancing angles. We show the difference between using Schlick's method with F_0 values calculated for red, green, and blue separately and the LSK method in Figures 10, 11, and 12 for gold, copper, and zinc. Their respective comparison with a spectral Fresnel is shown in the graphs, while we use the full Fresnel equation in the images.

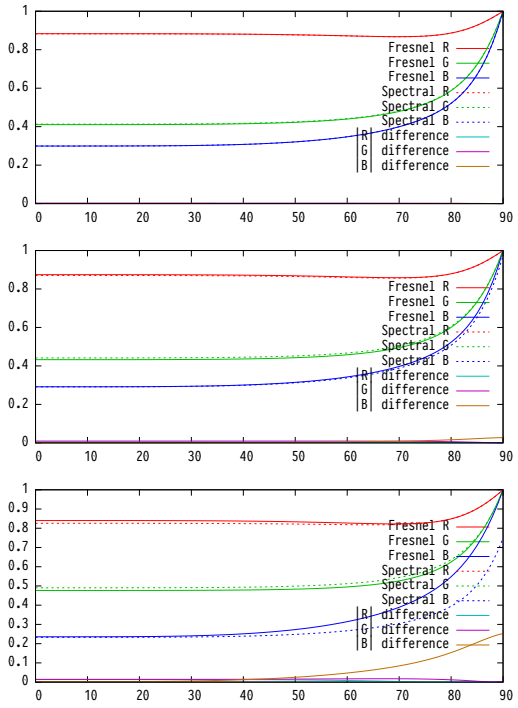


Figure 13: A comparison of setting the filter width to 12nm, 25nm, and 50nm. The error lines show increasing error as filter width increases. The material here is copper.

We find that our method breaks down when setting the filter too wide. Figure 13 shows how setting the filter wide results in greater angle towards incidence. The reason for this is that we are calculating too much blending between red, green, and blue values. If we were to continue widening the filter, we eventually will get a monochromatic reflection value.

7 CONCLUSION AND FUTURE WORK

We have shown that the display is an integral part of getting the Fresnel equations to work correctly for multispectral materials such as metals. We have demonstrated that the complex index of refraction can be sampled so that a spectral rendering is not required for realtime applications. We introduced a novel method of choosing appropriate wavelengths based on the chromaticity specification. We have also created a web based graphics experiment that uses a variety of different materials.

Our results do not have any impact on realtime frame rate performance, but they do enhance the quality of the image because the complex index of refraction is appropriately calculated for RGB color displays. We introduced a specular color method which could be suitable for applications with low CPU/GPU speeds. Finally, we surveyed the available Fresnel equations and

ISBN 978-80-86943-41-1

approximations and their suitability in terms of error to a spectral Fresnel equation implementation.

In the future, we hope to continue our work with multispectral materials. This includes work on dispersion and thin films. We also hope to experiment with anisotropic microfacet distribution models. We also plan to extend our sun and moon illumination model to accurately simulate astronomical phenomena to create accurate skylit illumination for both day and night to create realistic illumination estimates for future experiments (the current state of the art is daylight only).

8 REFERENCES

[Ado98] The adobe rgb (1998) color image encoding. Technical report, Adobe Systems Incorporated, 1998.

[AMHH08] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering, Third Edition*. CRC Press, 2008.

[Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.

[BS12] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7, 2012.

[Bul79] M.G. Bulmer. *Principles of Statistics*. Dover Books on Mathematics Series. Dover Publications, 1979.

[Com16] Wikimedia Commons. File:ciexy1931 rec 2020 and rec 709.svg — wikimedia commons, the free media repository, 2016. [Online; accessed 10-March-2018].

[CT81] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *SIGGRAPH Comput. Graph.*, 15(3):307–316, August 1981.

[fil18] Refractive index database. Technical report, Filmetrics, Inc., 3 2018. [Online; accessed 23-Jan-2018].

[Gla95] Andrew S Glassner. *Principles of Digital Image Synthesis*. Elsevier, 1995.

[Hei14] Eric Heitz. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT)*, 3(2):48–107, June 2014.

[IEC99] IEC. Multimedia systems and equipment - colour measurement and management - part 2-1: Colour management - default rgb colour space - srgb. Technical report, InternationalÂ ElectrotechnicalÂ Commission, 1999.

[JG78] George H. Joblove and Donald Greenberg. Color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, 12(3):20–25, August 1978.

[Kaj86] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 143–150, New York, NY, USA, 1986. ACM.

[LS05] István Lazányi and László Szirmay-Kalos. Fresnel term approximations for metals. In *The 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005, WSCG 2005, University of West Bohemia, Campus Bory, Plzen-Bory, Czech Republic, January 31 - February 4, 2005*, pages 77–80, 2005.

[MG80] Gary W. Meyer and Donald P. Greenberg. Perceptual color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, 14(3):254–261, July 1980.

[Mod13] Michael F. Modest. *Radiative Heat Transfer*. Academic Press, 3 edition, 2 2013.

[Nic65] Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.

[PH10] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.

[Sch94] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.

[SMP11] Rp 431-2:2011 - smpte recommended practice - d-cinema quality - reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

 ISBN 978-80-86943-41-1

Service-based Processing and Provisioning of Image-Abstraction Techniques

M. Richter[1] marvin.richter@hpi.de
M. Söchting[1] maximilian.soechting@student.hpi.de
A. Semmo[1] amir.semmo@hpi.de
J. Döllner[1] doellner@hpi.de
M. Trapp[1] trapp@hpi.de
[1]Hasso Plattner Institute, Faculty of Digital Engineering, University of Potsdam, Germany

ABSTRACT

Digital images and image streams represent two major categories of media captured, delivered, and shared on the Web. Techniques for their analysis, classification, and processing are fundamental building blocks in today's digital media applications ranging from mobile image transformation apps to professional digital production suites. To efficiently process such digital media (1) independent of hardware requirements, (2) at different data complexity scales, while (3) yielding high-quality results, poses several challenges for software frameworks and hardware systems, in particular for mobile devices. With respect to these aspects, using service-based architectures is a common approach to strive for. However, unlike geodata, there is currently no standard approach for service definition, implementation, and orchestration in the domain of digital images and videos. This paper presents an approach for service-based image-processing and provisioning of processing techniques by the example of image-abstraction techniques. The generality and feasibility of the proposed system is demonstrated by different client applications that have been implemented for the Android operating system, for Google's G-Suite Software-as-a-Service infrastructure, as well as for desktop systems. The performance of the system is discussed at the example of complex, resource-intensive image-abstraction techniques, such as watercolor rendering.

Keywords

Image-processing techniques, service-based processing, service-based provisioning

1 INTRODUCTION

For many years, digital images have been usually captured by means of mobile devices, such as smartphones and digital cameras, and have been widely distributed by a number of different media channels. Especially on the Web, with the emergence of social media and image platforms (e.g., Instagram), the amount of digital images increases dramatically. In these contexts, techniques for image analysis, classification, and processing are required, e.g., to optimize image search engines, to develop image transformation software for the application of image-abstraction techniques [12], and to produce artistic effects, such as style transfers [18].

With respect to image transformations on mobile devices, two technical alternatives are predominant: *on-device* and *off-device* processing. On-device processing utilizes the device hardware (e.g., CPU and GPU) to transform images and, thus, does not depend on a remote processing server [2], resulting in three significant benefits: (1) processing is usually faster, since there is no round-trip-time to a server (*performance*), (2) if images contain personal information, they are not intended for sharing via the network (*privacy*), and (3) although network communication has become prevalent, it is not guaranteed to establish network connections, which especially applies to mobile devices using mobile networks (*reliability*).

However, on-device image-processing has also a number of limitations. The design, implementation, testing, deployment, and maintenance of hardware-accelerated software for on-device processing is a cost-intensive process in terms of development-time and resources due to the high diversity of devices. A multitude of platforms (e.g., iOS, Android, Windows, macOS, Linux) with different graphics Application Programming Interfaces (APIs) (e.g., OpenGL, OpenCL, Direct3D, CUDA) in different versions has to be supported.

In particular, performing image-processing tasks on mobile devices have the following implications: on the mobile Android market a huge device heterogeneity is ubiquitous, often implying different capabilities and hardware specifications. Technical evaluations of an on-device image-processing approach on Android devices revealed huge performance differences between

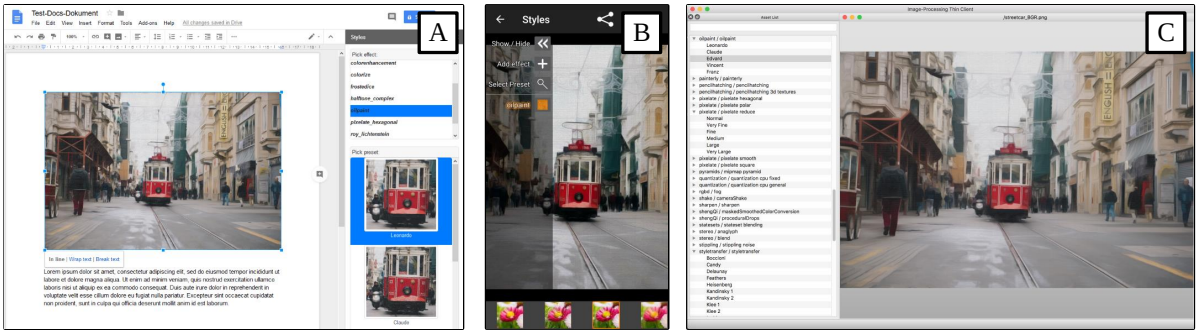


Figure 1: Application of an image-abstraction technique (oil paint) to the same image on different platforms: Web-based Google Office Suite (A), mobile Android app (B), and desktop client (C).

devices with similar hardware specifications [2]. Also, mobile devices are limited with respect to memory available, computational power, and battery life, which is a crucial limitation, since image-processing is usually an intense resource consuming process. Further, software developer have to take various inputs via device sensors as well as device orientation and display resolutions into account. Moreover, small displays often lead to screen real estate problems. Furthermore, Digital Rights Management (DRM) for on-device solutions cannot be guaranteed, since processing relevant data (e.g., textures, shaders) is accessible via API-call interceptions and, thus, the control and protection of intellectual property gets lost.

Service-based Image-Processing.

With respect to the on-device image-processing challenges stated above, an off-device service-based image-processing approach offers a practical alternative. For this purpose a Service-Oriented Architecture (SOA) and its predominant design patterns is used [3]. It facilitates development of reusable components for processing images without the downsides of dealing with a wide range of devices. This allows a simple integration of those components into existing services. Thus, various heterogeneous clients can be attached to provide hardware independent cross-platform solutions via a common Representational State Transfer (REST) interface [6]. For example, Figure 1 shows the application of a complex oilpaint effect [19], a typical resource-intensive artistic stylization technique, on different platforms, such as web-based Google Office Suite add-ons, mobile Android apps, and desktop applications that interact with the proposed service-based image-processing server.

The approach of service-based processing offloads computation-intensive image-processing tasks (e.g., processing of stylization effects that rely on multi-stage processes) to a server infrastructure, equipped with more resources in terms of memory, computation power, and energy supply. Thus, this approach dramatically reduces energy consumption especially for mobile devices. Further, an important requirement for future image transformation processes is the capability to deal with continuously increasing *spatial* and *temporal* resolution of visual media (>20 megapixel at 120 frames-per-second) and to guarantee high quality output images. A service-based approach tackles the complexity of scalable and high-resolution image space. The applicability of the service-based approach and its performance highly relies on the bandwidth of the used network for data transmission. However, due to constantly evolving network infrastructure in terms of increasing bandwidth, it is an acceptable constraint.

Contributions. To summarize, this paper makes the following contributions:

1. It presents a concept for service-based off-device image-processing, following the orchestration pattern for service composition. Exemplarily, the creation of a high-level service based on atomic low-level services is shown as an example.

2. A platform-independent representation of image-processing effects is formulated for highly customizable configuration of the image-processing. Furthermore, a approach for storing and provisioning of these effect representations is presented.

3. It demonstrates the approach using different application examples, such as image manipulation on Android and desktop systems, and the integration into the Google Office Suite for various image-abstraction techniques (e.g., oilpaint, watercolor).

The remainder of this paper is structured as follows. Section 2 reviews related work on service-based approaches to image-processing and applications. Section 3 presents our concept for service-based provisioning of image-abstraction techniques and explains implementation details. Section 4 shows common application examples and Section 5 discusses our approach and states potential future research. Finally, Section 6 concludes this paper.

ISBN 978-80-86943-41-1

2 RELATED WORK

This section covers related work in the field of service-based image-processing and approaches for image-abstraction techniques.

Service-based Image-Processing. Several software architectural patterns are feasible for implementing service-based image-processing. One prominent style of building a web-based processing system for any data is the service-oriented architecture [22]. This approach allows server developers to set up a multitude of processing endpoints, each providing a specific functionality and covering a different use case. These endpoints appear as a single entity to the client, i.e. the implementation stays hidden for the requesting clients, but can be realised through an arbitrary number of self-contained services. This work follows the service-oriented architecture as described in Section 3.

Since web services are usually designed to maximize their reusability, their functionality should be simple and atomic. Therefore, the composition of services is critical for fulfilling more complex use cases [14]. The two most promiment patterns for realising this composition are choreography and orchestration. The choreography pattern describes decentralized collaboration directly between modules without a central component. The orchestration pattern describes collaboration through a central module, which triggers the different web services and passes the intermediate results between them. In this work, the orchestration pattern is implemented as described in Section 3.

In the field of image analysis, Wursch et al. [26] developed a web-based tool that enables users to perform different image analysis methods, such as text line extraction, binarization, and layout analysis. The tool is realised through a set of REST web services. Application examples in that work include multiple web-based applications for different use cases.

The viability of implementing image-processing web services using REST has been demonstrated by Winkler et al. [24], including the ease of combination of endpoints. Another example for service-based image-processing is Leadtools (https://www.leadtools.com), which provides a fixed set of around 200 image-processing functions with a fixed configuration set via a web API. In this work, however, a similar approach using REST is chosen, although with a different focus in terms of granularity of services. While Winkler and Leadtools focus on fixed endpoints for a selected number of image-processing effects, this work aims for a general-purpose image-processing system based on an platform-indepedent effect format (Subsection 3.1).

In the field of geodata, the Open Geospatial Consortium (OGC) set standards for a complete server-client ecosystem. As part of this specification, different web services for geodata are introduced [15]. Each web service is defined through specific input and output data and the ability to self-describe its functionality. In contrast, in the domain of general image-processing there is no such standardization yet. However, it is possible to transfer concepts from the OGC standard, such as unified data models. These data models are realised through a platform-independent effect format. In the future, it is possible to transfer even more concepts set by the OGC to the general image-processing domain, such as the standardized self-description of services.

Image-Abstraction Techniques. In this work, we focus on edge-aware and content-preserving image-processing as a fundamental tool in computational photography and non-photorealistic rendering for abstraction and artistic stylization. Typical approaches that operate in the spatial domain for abstraction use a kind of anisotropic diffusion [17, 23] and are designed for parallel execution, such as approximated by the bilateral filter [21] and guided filter [9]. A plentitude of stylization techniques exist using these filters as building blocks to simulate traditional painting media and effects [13], such as cartoon [25] and oil paint [20]. However, these may become computationally expensive when applied in an iterative multi-stage process. This particularly applies to techniques using global optimizations to separate detail from base information, e.g., based on weighted least squares [4] or locally weighted histograms [11], and recent techniques that separate style from content using neural networks [7]. Because of their global optimization scheme, they are typically not suited for real-time application, in particular not on mobile devices. To this end, we implemented a variety of these techniques using the proposed image-processing service including stylization, HDR tone mapping and compression, JPEG artifact removal and colorization, to demonstrate its versatile application.

3 SERVICE-BASED PROCESSING

Figure 2 shows a conceptual overview of the components of the image-processing system. It basically comprises the following components, which are described in the remainder of this section in greater detail:

Effect Service: This service component is responsible for storing all resources required by the image-processing service. It delivers platform-independent representations of image-processing effects based on an Extensible Markup Language (XML) format bundled with Graphics Processing Unit (GPU) shader programs and textures for dedicated target platforms, i.e., GPU hardware and API. This service can be utilized for different use cases in addition to the one described in this work, e.g., delivering platform-independent image-processing effects directly to user clients for on-device rendering [2].

ISBN 978-80-86943-41-1

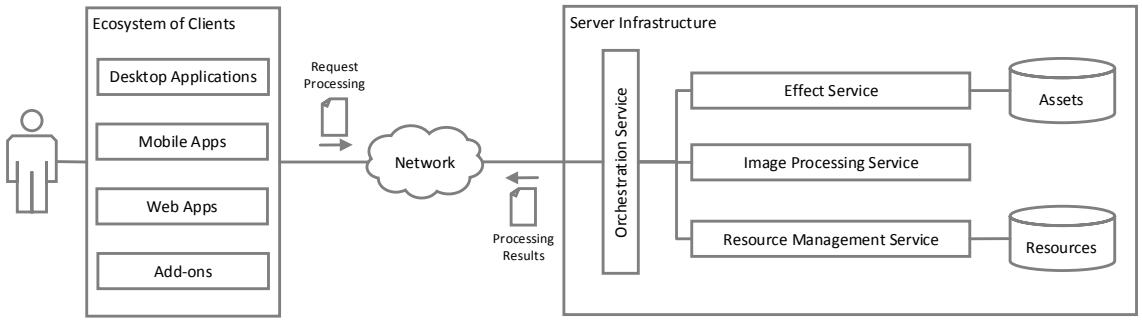


Figure 2: Components of the proposed service-based image-processing system. A diverse range of clients can request the image-processing functionality via the interface of the orchestration service, which coordinates services for effect provisioning (effect service), image-abstraction and stylization (image-processing service), and storing the respective data, e.g., images, (resource management service).

Image-Processing Service: This service performs the actual processing of images with respect to image analysis, abstraction, and stylization. It receives an input image, an effect reference, and a processing configuration (e.g., output file format, output quality etc.) and delivers a processing result, which, for example, can be a stylized image or an analysis result (e.g., dominant color, histogram).

Resource Management Service: The resource management service is responsible for storing and provision of data, such as images and its metadata. Along with the provision of resources this service also provides low-level image manipulation functionality, such as cropping, resizing, and rotating.

Orchestration Service: A service-based system consisting of various service components has to tackle the crucial challenge of service composition. The composition of services aims at creating a composite service that combines various atomic functional building blocks provided by the available services in order to satisfy a higher level use case. The composition is managed by the orchestration service. Further, the orchestration service provides the public service endpoints that can be accessed by the clients.

3.1 Effect Representation and Service

The effect service is responsible for delivering platform-independent descriptions of image-processing effects to requesting clients, such as the image-processing service. The delivered image-processing effects are composed of multiple assets, which depend on each other. Each asset is described in an asset format. This asset format and the dependency structure is based on the work of Duerschmid et al. [2]. Examples for assets are implementation-specific files such as shaders and textures. The effect service is realised through a Node.js JavaScript application, which provides the web service interface, in conjunction with PostgreSQL, which is used for storing asset meta data and asset files.

Assets are sets of files that, once composed, define a platform-independent, executable description of an image-processing effect. Assets are designed to strictly separate between platform-independent parts and platform-dependent parts. The composition of assets is realised through inter-asset dependencies. Once an image-processing operation is requested, the effect service resolves the dependencies and bundles all assets together, resulting in an executable asset bundle. This minimizes the required client effort to download a given image-processing effect.

3.2 Image-Processing Service

The image-processing service is responsible for executing image-processing effects. The service implementation uses a cross-platform C++ image and video processing framework designed for desktop and server systems. To enable efficient processing, graphic acceleration supporting multiple modern graphic APIs (e.g., OpenGL, Vulkan) is used. Per request, the image-processing service takes the reference to an image as input and can be configured with respect to the following configurations:

Effect File: Reference to the parameters of the desired effect. The effect's XML file is parsed and a processing pipeline is instantiated based on the respective effect configuration.

Preset Identifier: The effect can be configured using so-called presets. Setting a preset identifier applies a predefined parameter configuration that is basically a list of parameter name and value tuples.

Output File Format: To reduce streaming bandwidth and the size of transmitted data, the image-processing service can generate output images in

 ISBN 978-80-86943-41-1

compressed, lossy formats (e.g., JPEG) and lossless formats (e.g., PNG). The compression setting can be configured by the client according to the specific use case.

Output File Quality: The quality parameter configures the JPEG quality and the PNG compression rate, respectively. The quality factor of the output image must be in the range 0 to 100. As for PNG export, specify 0 to obtain small compressed files and 100 for large uncompressed files. The format and the respective quality configurations results from a trade-off between output quality and transmission time. Those are crucial adjustments to achieve a responsive user interaction within the client application.

Return Type: The return type determines, if an *image token* or the processed *image* is returned by the service. Choosing an image token reduces bandwidth while applying multiple effects and/or different effect presets, since the image is kept on the server and does not need to be transmitted for every single request.

3.3 Resource Management Service

Two types of resources are associated with the resource management service: *image* and *image-related* resources. Supported image file formats are the ubiquitous MIME types JPEG and PNG. The support for further file formats can be easily integrated. Every image resource can be identified using a Universal Resource Identifier (URI). The image information includes both technical properties about the image (e.g., image resolution) and low-level analysis results such as pixel edge-count, dominant colors, or a color histogram. In addition to the management of resources and their metadata, the resource management service provides low-level image manipulation functionality (e.g., rotating, cropping, resizing), which can be used by image-viewer clients that require zoom, pan, or rotate functionality.

3.4 Orchestration Service

The orchestration service follows the facade design pattern—in the context of service-based architectures also denoted as API gateway—and provides the public service-endpoints that handles requests sent by the clients. Requests that require atomic functionality, which is provided by a single service, are dispatched to the respective service (e.g., provision of an image, update of an effect). More complex requests (e.g., processing of an image with a specific effect) that depend on several services to fulfill a high-level use cases are managed by the orchestration service. The orchestration service calls and coordinates required services, manages the execution flow, and assembles information required for the response. A workflow example for processing an image that uses the orchestration of the three main services, is outlined in the following:

1. Fetch image from resource management service.
2. Fetch the requested effect from the effect service.
3. Request the processing of the fetched image with the requested effect at the image-processing service.
4. Gather and deliver relevant status/error information and the processing results.

Further, the orchestration service delivers its capabilities using the OpenAPI specification (https://www.openapis.org/). Thus, the service endpoints are both human and machine readable and can be comprehended without reading source code or documentations.

In technical terms, communication between the services is based on RESTful HTTP [5] – which is one of the established standards, along with SOAP [8] and message oriented middleware [1] – for service-based architectures [16]. REST facilitates the communication within heterogeneous environments comprising services running on different machines or in different execution environments. The REST services accept HTTP requests to a URI with a specific HTTP method (PUT, GET, POST, DELETE). The URI for requesting the orchestration service complies the following template:

```
https://<IP>/<resource>/<id>/<action>
```

Here, the parameter `<IP>` refers to the location of the server, `<resource>` indicates the type of the resource (e.g., image), `<id>` specifies the identifier of the resource, and `<action>` declares the type of processing, (e.g., transform, analysis, or info). Additionally, HTTP methods are used to map CRUD (create, read, update, delete) operations to HTTP requests. Hence, a resource can be created or updated (POST), fetched (GET), and deleted (DELETE). To get or delete a resource, the action path parameter can be omitted. A request to retrieve the image information of a specific image with the identifier 12345 is the following GET request:

```
https://<IP>/image/12345/info
```

4 APPLICATION EXAMPLES

This section demonstrates the applicability of the presented concept to various application domains, such as (web-based) add-ons for office products (Subsection 4.1), image manipulation on mobile devices (Subsection 4.2), and desktop systems (Subsection 4.3).

 ISBN 978-80-86943-41-1

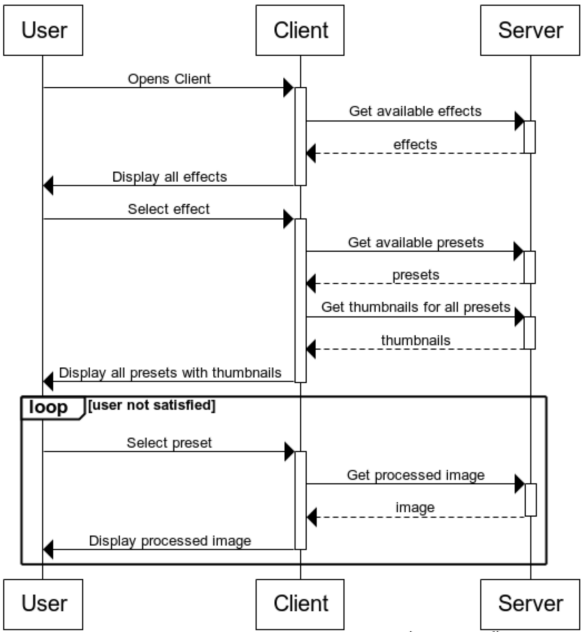


Figure 3: The common workflow of all clients communicating with the web services. From top to bottom: Initially, all effects are retrieved and displayed to the user. The user then has the option to try out different effect and preset combinations until he is satisfied.

4.1 Google Office Suite Integration

Google offers various web apps as part of its office suite (G-Suite): Sheets, Docs and Slides. Each of these Google Apps offers add-on integration through the Google Apps Script platform. As a demonstration for the integration of service-based image-abstraction techniques, add-ons for Google Sheets, Docs and Slides that utilize the presented image-processing web service of this work were developed (Figure 4).

The workflow of the add-on, which is common for all application examples of this chapter, is shown in Figure 3. First, the client requests a list of all currently

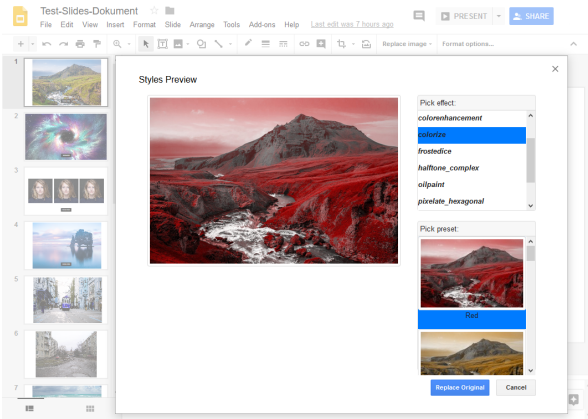


Figure 4: Example of the integration of service-based image-abstraction techniques into Google Slides via an Add-on using server-sided Google Script.

available effects from the effect service. These are displayed to the user in a list. Once the user selects an effect, the currently selected image is cropped and processed multiple times to generate previews for the different effect presets. These dynamic previews are displayed below the effect list, enabling users to get a first impression of the visual impact of the effects and its respective presets. Subsequent to selecting one of the previews, a full-resolution image is processed on the server and displayed to the user. This full-resolution image can then be inserted into the Google Slides, Docs, or Sheets document.

The add-ons are implemented using Google Script, a server-side scripting interface based on JavaScript, and templated HTML5 user interfaces. Figure 5 shows an overview of the basic add-on architecture and integration. Since every Google App has a different API for retrieving images from the document, each add-on requires to be adapted for the specific app. The core of all add-ons, the web-service connection library, encapsulates common functions required for calling the web services. Using this design, rapid development of add-ons for new and existing Google Apps is easily possible.

However, the add-on environment of Google Apps Script poses two limitations. First, the daily quota for regular users of HTTP requests is limited to 100 megabytes. A naive implementation that directly uploads and downloads processed images can reach this quota quickly. To counter-balance this, the add-on is designed to send links instead and exploit HTML image-embeddings whenever possible to circumvent the quota. A second limitation currently concerns the runtime performance of inserting new images into a document. In a Google Slides presentation example, inserting a 1.6 megapixel image takes 3.5 seconds. In contrast, uploading, processing and downloading the image takes only approx. 0.8 seconds in the same environment. Therefore, to achieve sufficient performance, the add-on is implemented to minimize these calls to Google APIs.

4.2 Android Image Manipulation App

Figure 6 shows screenshots from an Android mobile app that enables the application of various image-processing techniques to input images. This client offers similar features as ProsumerFX [2] but without on-device processing.

The workflow in this app is similar to the presented Google Office Suite add-on. At first, the client fetches a list of available image-processing techniques from the web services. Next, these effects are filtered and grouped based on delivered effect metadata, such as effect complexity and category. Once the user decides for an effect, dynamic previews for each predefined parameter configuration (preset) via thumbnail processing are

ISBN 978-80-86943-41-1

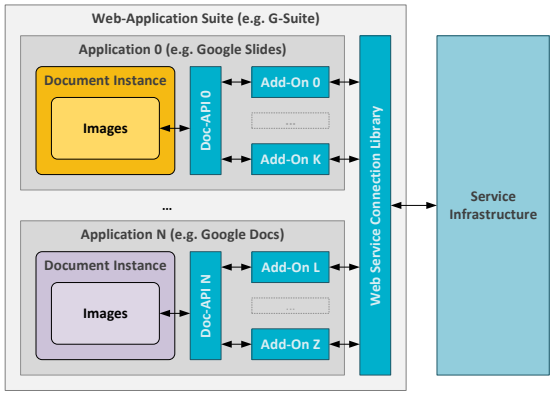


Figure 5: Architecture of the Google Office Suite add-ons. For every Google App, an arbitrary number of add-ons with a different set of available effects can be deployed. All add-ons use the web service connection library, which provides convenient access to the web services.

generated. This allows users to get a first impression of the visual impact of each preset and helps them in their decision making. When the user selects a preset, a high-quality version of the processed image will be generated on the server and then displayed in the client. The app also allows the combination of multiple effects, sequentially applying them to the image. The user can reorder these effects, taking control over the orchestration of the web services.

There are multiple advantages of the app in contrast to on-device processing. First, the processing is independent of the device graphics hardware since it is performed on a server machine. This allows weaker devices to apply complex image-processing effects to high-resolution images. Furthermore, the battery consumption is significantly lower than a comparable on-device rendering solution. The app can also be utilized in a business context as a white-label solution for different companies, i.e., multiple customized versions of the app with a specific brand, logo, and identity can be created and sold to companies.

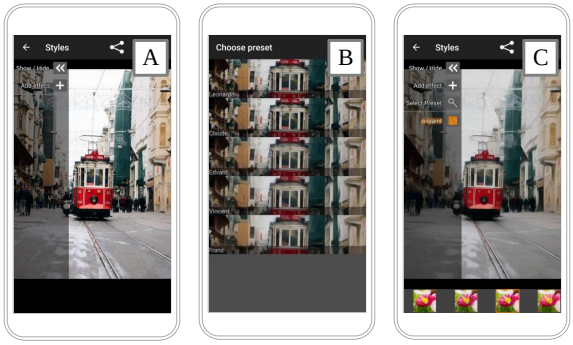


Figure 6: Workflow of the Android image manipulation app. After selecting an image (A) the user can choose an effect and one of the predefined parameter configurations (B) and apply them to the input image (C).

4.3 Desktop Client

The desktop client is realised through a C++ framework, using the Qt application framework that communicates with the orchestration service to utilize the functionality of the server component and to fulfill high-level use cases. The presented approach is tested using a Command-Line Interface (CLI) and a Graphical User-Interface (GUI) application.

CLI Application. The CLI application is used as a rapid application development framework for the analysis and processing of images and image collections. Further, it provides statistics and reports of effects and additional metadata. The CLI can be easily utilized by developers or desktop applications to implement more complex functionality based on the provided effects. For instance, it can be used as a convenient tool to test effects and retrieve an overview of the effect status (e.g., average runtime, load, and bandwidth tests as well as errors). Further, it supports both basic low-level functionality (e.g., listing available effects and filter them) and more advanced features (e.g., batch processing images that reside in specific folders or whose filenames comply to a specified regular expression).

GUI Application. The GUI application demonstrates the applicability of a simple cross-platform image-processing app that applies selected effects with specified presets subsequently to an image. The application is implemented with the Qt GUI module that facilitates the development and deployment of cross-platform software for various desktop systems (Figure 1-C). Because of the minimalistic user interface, the application is well-suited for casual non-professional users, who want to import, process, and export a single image.

5 RESULTS AND DISCUSSION

This section discusses the results obtained by our application examples with respect to its runtime performance, current limitations of the presented approach, and future research questions to address.

5.1 Performance Observations

In general, two major factors affect the runtime performance of service-based architectures: *data transmission* and *data processing*. The transmission time of input and output images over a network highly depends on the image data size and the available network bandwidth. Timings of data transmission are not examined in this paper. The image-processing service output configurations via the format and quality parameters account

ISBN 978-80-86943-41-1

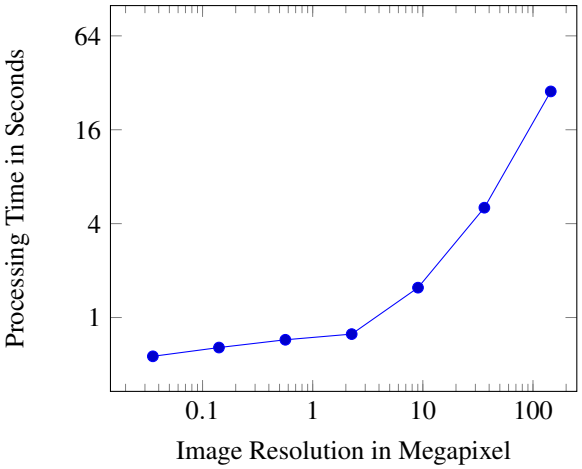


Figure 7: Performance of the image-processing service applying the watercolor effect on images with different resolutions (0.01 to 145 megapixels).

that. For example, within the Google Office Suite add-on, a low-quality image with high compression can be chosen for preview images. The machine hosting a single instance of the processing service has the following specification: CPU: Intel(R) Xeon(R) CPU E5-2637 v4 @ 3.50GHz; GPU: NVIDIA(R) Quadro M6000 24GB; RAM: 64.00 GB; Hard drive: 4 × SanDisk X400 2.5 512 GB in RAID 0; OS: Ubuntu 16.04.3 LTS 64-bit.

The performance measurements of the image-processing service (Figure 7) allow the following observations: (1) the runtime depends linearly on the resolution of the input image while for common resolutions the processing achieves acceptable performance (200-500ms), (2) high-resolution image data up to 16.384×16.384 pixels can be processed, but requires approx. 15 GB RAM; an amount that can hardly be managed during on-device processing on mobile devices, and (3) the prototypical implementation of the image-processing service is hardly real-time capable for common HD resolutions and, thus, is not suited for video-stream processing yet.

Table 1 shows the runtime performance of the image-processing service for different effects with respect to the major processing stages:

Effect Loading: Load the effect, which includes parsing of the XML representation and instantiation of the effect pipeline with the associated GPU objects.

Image Decoding: Load the image, allocate texture memory and data transfer for the image.

Image Processing: Execute the effect pipeline. The result will be stored in the pipeline output texture.

Image Encoding: Export the image to a specified format (e.g., JPEG and PNG) with a specified compression strategy and write it back to disk.

Table 1: Runtime performance of image-processing stages using effects of different complexity. The input is a 2 megapixel JPEG image.

Effect	Load Effect	Decoding	Processing	Encoding	Total
Blur	111ms	169ms	20ms	17ms	**317ms**
Emboss	157ms	174ms	24ms	25ms	**380ms**
LUT	196ms	164ms	25ms	23ms	**408ms**
Oilpaint	242ms	170ms	47ms	25ms	**484ms**
Watercolor	523ms	163ms	67ms	30ms	**783ms**

On complex effects (e.g., oilpaint, watercolor) loading of effects and processing requires more time than on simple effects (e.g., emboss and blur filter). Compared to the on-device equivalent the effect loading stage, which takes 1-3 seconds on device, is significantly faster [2]. However, input and output operations (effect fetching, decoding, encoding) poses the bottleneck of the processing component. Caching mechanisms for the effect fetching stage can result in a considerable speedup, while on high-resolution images the decoding and encoding stages become the major time consuming parts.

5.2 Limitations

The prototypical implementation of the presented approach exhibits some limitations. Since the reusability of web services is maximised, each single service often only provides simple and atomic functionality. Therefore, to enable more complex use cases, the composition of services is a critical question to address. Previous research has shown that composition can be performed automatically, once the web service ecosystem and the desired use cases are strictly formalized [10]. In this work, the composition of services was implemented using a meta service, i.e., the orchestration service described in Subsection 3.4, and client-side, i.e., the thumbnail generation shown in Subsection 4.1 and the effect composition described in Subsection 4.2. As described in the according sections, these approaches come with their own limitations respectively.

Furthermore, the current prototypical implementation assumes a monolithic image-processing service. With the assumption of this service only being handled by one physical machine, this could turn out to be a limiting factor for the scalability of the complete system. A load-balancing system in addition to multiple server-instances would be a potential improvement to overcome this limitation. In addition thereto, each server-instance is still limited with respect to processing capabilities such as maximum texture sizes, number of shader cores, or memory bandwidth.

5.3 Future Research Directions

On the basis of the presented results, various future research directions are possible. Extending the web

ISBN 978-80-86943-41-1

service processing approach with capabilities for video processing is planned for future work. Here, a possible implementation for efficient real-time video processing could involve using live streaming protocols, e.g., Real-time Streaming Protocol (RTSP), and exploiting compression methods. Another aspect to improve the performance of the presented system is scalability. In addition to introducing load balancing in front of the image-processing server(s), the image-processor can be extended to support tiled rendering. This would allow the system to process even higher resolutions than 16.384×16.384 pixels, which might facilitate more application fields for this work, such as the geodata domain.

Furthermore, support of an extended effect parameterization beyond choosing presets might be desirable to some users or application integrations. Allowing fine-grained control over every effect parameter increases the creative freedom but might make it harder for users to achieve desirable results. Featuring a service-oriented architecture, the parameters could be exposed via self-description of services. Furthermore, allowing users to share their own parameterizations as new effects on a platform as demonstrated in [2] is imaginable.

6 CONCLUSIONS

This paper presents a novel concept for service-based processing and provisioning of image-processing techniques with respect to extensibility and applicability of image effects to further domains and current software and hardware systems. The approach is based on the design of atomic services that are orchestrated to higher level services to fulfil sophisticated use cases, such as applying configurable image-abstraction effects to images. Since the image-processing is performed on the server, clients are not responsible for resource-intensive processing tasks and the image-processing can be implemented and optimized only once for a known GPU environment. The presented approach enables cross-platform interoperability with a diverse range of heterogeneous clients. The applicability of the approach is demonstrated to various application domains, such as mobile and desktop applications. The image-abstraction effects are described via a platform-independent representation that enables a highly customizable configuration. These effects and their dependent assets are stored on the server, which provides a high degree of protection for sensitive data (e.g., intellectual property). As part of the technical evaluation performance measurements showed the general applicability of the approach, i.e., image-processing can be performed in a reasonable amount of time. Also, the processing and output of high quality images with resolutions up to 145 megapixels have been shown.

ACKNOWLEDGMENTS

We thank Markus Brand, Merlin de la Haye, Phaedra Goudoulaki, Erik Griese, Moritz Hilscher, Alexander Riese, and Hendrik Tjabben for their contributions to the design and implementation of the presented system. This work was partly funded by the Federal Ministry of Education and Research (BMBF), Germany, for the AVA project 01IS15041B. We further thank Digital Masterpieces GmbH for providing data sets.

REFERENCES

[1] Edward Curry. Message-Oriented Middleware. In Qusay H Mahmoud, editor, *Middleware for Communications*, chapter 1, pages 1–28. John Wiley and Sons, Chichester, England, 2004.

[2] Tobias Dürschmid, Maximilian Söchting, Amir Semmo, Matthias Trapp, and Jürgen Döllner. Prosumerfx: Mobile design of image stylization components. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pages 1:1–1:8, New York, NY, USA, 2017. ACM.

[3] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

[4] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics*, 27(3):67:1–67:10, 2008.

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Rfc 2616, hypertext transfer protocol – http/1.1, 1999.

[6] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. AAI9980887.

[7] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, Los Alamitos, 2016. IEEE Computer Society.

[8] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon. Soap version 1.2 part 1: Messaging framework (second edition). World Wide Web Consortium, Recommendation REC-soap12-part1-20070427, 2007.

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *Proc. European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2010.

[10] Alexander Jungmann and Bernd Kleinjohann. Automatic composition of service-based image

 ISBN 978-80-86943-41-1

processing applications. In *Proc. IEEE International Conference on Services Computing (SCC)*, pages 106–113. IEEE Computer Society, 2016.

[11] Michael Kass and Justin Solomon. Smoothed local histogram filters. *ACM Transactions on Graphics*, 29(4):100:1–100:10, 2010.

[12] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2013.

[13] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2013.

[14] Angel Lagares Lemos, Florian Daniel, and Boualem Benatallah. Web service composition: A survey of techniques and tools. *ACM Computing Surveys*, 48(3):33:1–33:41, 2015.

[15] Matthias Mueller and Benjamin Pross. *OGC® WPS 2.0.2 Interface Standard*. Open Geospatial Consortium, 2015. http://docs.opengeospatial.org/is/14-065/14-065.html.

[16] Mike P. Papazoglou and Willem-Jan Heuvel. Service oriented architectures: Approaches, technologies and research issues. *The VLDB Journal*, 16(3):389–415, 2007.

[17] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[18] Amir Semmo, Tobias Isenberg, and Jürgen Döllner. Neural style transfer: A paradigm shift for image-based artistic rendering? Proceedings International Symposium on Non-Photorealistic Animation and Rendering (NPAR), pages 5:1–5:13, New York, 7 2017. ACM.

[19] Amir Semmo, Daniel Limberger, Jan Eric Kyprianidis, and Jürgen Döllner. Image Stylization by Interactive Oil Paint Filtering. *Computers & Graphics*, 55:157–171, 2016.

[20] Amir Semmo, Daniel Limberger, Jan Eric Kyprianidis, and Jürgen Döllner. Image stylization by interactive oil paint filtering. *Computers & Graphics*, 55:157–171, 2016.

[21] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proc. International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE, 1998.

[22] Mircea-Florin Vaida, Valeriu Todica, and Marcel Cremene. Service oriented architecture for medical image processing. *International Journal of Computer Assisted Radiology and Surgery*, 3(3):363–369, 2008.

[23] Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[24] Robert P. Winkler and Chris Schlesiger. Image processing rest web services. Technical Report ARL-TR-6393, Army Research Laboraty, Adelphi, MD 20783-119, 2013.

[25] Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226, 2006.

[26] Marcel Würsch, Rolf Ingold, and Marcus Liwicki. Divaservices - a restful web service for document image analysis methods. *Digital Scholarship in the Humanities*, 32(1):i150–i156, 2017.

 ISBN 978-80-86943-41-1

Combination of Temporal Neural Networks for Improved Hand Gesture Classification

Aditya Tewari [#‡]
Aditya.Tewari@dfki.de

Bertram Taetz[‡]
Bertram.Taetz@dfki.de

Frédéric Grandidier[†]
Frederic.Grandidier@iee.lu

Didier Stricker[*]
Didier.Stricker@dfki.de

ABSTRACT

Low latency detection of human-machine interactions is an important problem. This work proposes faster detection of gestures using a combination of temporal features learnt on block time input and those learnt by contextual information. The results are reported on a standard in-car hand gesture classification challenge dataset. The recurrent neural networks which learn sequential contexts are combined with 3D convolutional neural networks (C3D). We have demonstrated that a design similar to various multi-column networks, which have been successful for image classification and understanding can also improve classification performance on varying length time series. Therefore, a combination of C3D and Long-Short-Term Memory (LSTM) is utilized for classification of hand gestures. On the task of early hand gesture classification, the proposed model outperforms the the C3D model which reports best results on full gestures. It is second best and only slightly less accurate than the best performing method, on the full gesture length.

Keywords

LSTM, 3D Convolution, Neural Network, Temporal Features, Hand Gestures, Automobile Application.

1 INTRODUCTION

One of the principles for the interaction system is the need for a short time delay between the start of the interaction and response from the machine [RSP11]. A low latency system is easier to use and is often essential. Gesture recognition systems inside cars use sensors that require low power expenditure. These cameras introduce an integration time versus frame-rate trade-off [GYB04]. This reduces the available frame rate, thus a solution where robust predictions are made on short length gesture videos is important in such situations. Therefore, not only classification of complete fast gestures, but also classification of slow but incomplete gestures is of interest. In this work we solve the problem of robust and fast classification of (incomplete) hand gestures.

Several methods for vision based HGR have been employed. Hand crafted features [TTGS16] containing temporal and spatial information have been regularly used. Hidden Markov Models (HMM) [CFH03] and Support Vector Machines (SVM) [LGS08] have been used for classification of spatio-temporal features. Other solutions use an articulated model of the hand and its deformation for gesture classification [KKKA13]. It was empirically demonstrated by [AAGES10] that among machine learning methods, neural network models, like multi-layer-perceptrons (MLP), are conducive to early predictions in time series data. The Recurrent Neural Networks (RNNs) were used for gesture classification by [YH15]. The work [OBT14] and [WKSL13] reported results on the challenging VIVA dataset, the performance of these methods was overtaken by neural network based methods proposed in [MGKK15].

It has been demonstrated that multi-modal approaches [OBT14, WKSL13] which employ trajectory shapes, boundaries and motion structures combination in a bag of features approach work better than single information approach. The positive influence of mutually independent information contributing to learning have been demonstrated in such works. This approach has also shown to have worked with neural networks. A C3D network [MGKP15] was

[*] DFKI, Augmented Vision, Trippstadter Str., 67663 Kaiserslautern, Germany.
[†] IEE-SA, Weiergewan, 11-rue Edmond Reuter, 5326 Contern, Luxembourg.
[‡] TU Kaiserslautern, Erwin-Schrodinger-Strasse 1, 67663 Kaiserslautern,Germany.

trained with data from multiple vision sensors and radar for better hand gesture predictions. The neural network solutions that use Multi-information methods use parallel neural networks, these networks have performed better than single column networks and have been used in various image classification tasks [CMS12]. The performance of activity recognition [DAHG+15] and hand gesture recognition have shown to improve by using a combination of parallel networks that accepts distinct data [MGKK15, CLS15].

1.1 Contributions and structure of the paper

It is of interest to investigate if the combination of features learnt from the same dataset but using distinct learning policies, thus resulting in dissimilar patterns, can contribute towards improving the learning performance. This investigation is inspired from the improvement in performance of multi-modal network when data with separate properties is used at the various input layers. To improve classification over time by using dis-similar concepts, we create multi-column networks with columns constituted from different temporal layers. A typical model that we propose has a parallel columns with one or more volumetric convolution layer and one column with recurrent layers.
We introduce a hand gesture recognition system that uses a combination of C3D and LSTM for identifying gestures at different delays from the start of gestures. This work combines the ideas of [MGKK15] with those of [DAHG+15].
Results on the VIVA challenge dataset [viv], which is a hand gesture classification dataset recorded on varying lighting conditions inside a car, are demonstrated. On a half length, incomplete gesture sequences, our proposed network outperforms the two column C3D model by a large margin of approximately 10%. Improvement in performance is noticed and reported on short incomplete sequences of gestures. The combination model performs better on half and quarter length incomplete gesture sequence.
In this paper we propose a new neural network configurations that improves the classification performance for short sequence gestures. To the best of our knowledge, this is the first effort to utilize the combination of the two temporal neural networks to make early classification of a time series signal. Gesture sequences with full length (32 frames), half length (16 frames) and quarter length (8 frames) from the beginning of the gestures are trained and tested for these architectures. The contributions can be summarized as:

- Introducing a method for improving the accuracy of early response in a gesture recognition system.

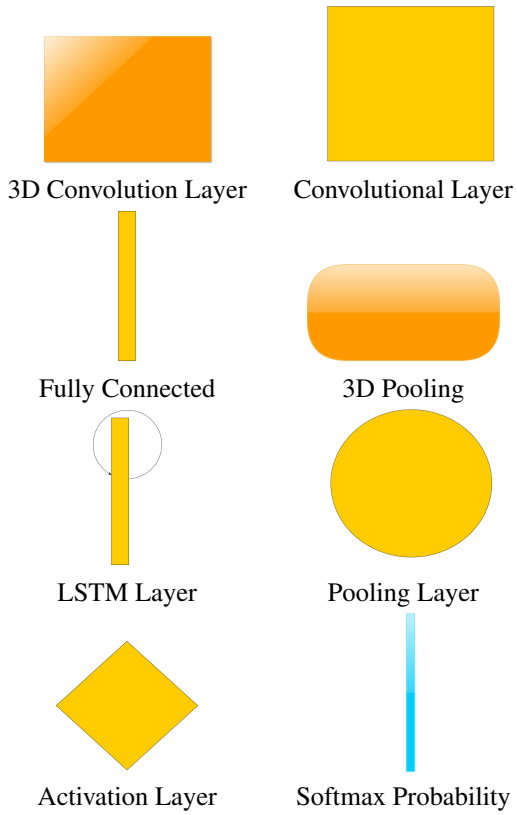


Figure 1: Labels for layer images used in this work

- Introducing a combination of block and context classification in time series by combining LSTM and C3D.

- An extensive analysis of various temporal neural network models for low latency classification of hand gestures.

In the Section 2 the dataset for the gesture experiments is explained, the sampling strategy and the preprocessing required to complete all the comparisons are described. The section Section 2.1 shows the training parameters and scheme used for various neural networks trained during the experiments. The Section 3 presents the C3D neural network that we later compare our proposed combination with. A benchmark is set in this section and test are also conducted with a multicolumn LSTM network. In the Section 4 we first train and validate the LSTM and C3D network and then propose a combination for better performance. These experiments are compared and the better performance of the combination network is demonstrated in the Section 6. The Section 7 presents an experiment on smaller models. Finally, in the Section 7.1, the possible extensions of this work and limitations are mentioned.

ISBN 978-80-86943-41-1

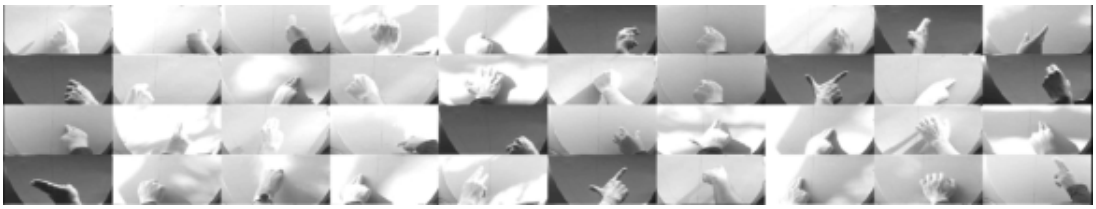


Figure 2: Representative Hand Gesture Dataset

2 DATA, SAMPLING AND TRAINING

The VIVA challenge dataset was used for these experiments. The gestures are defined by moving hands and changing or constant hand shapes. The VIVA challenge dataset has video sequences of fifteen hand gestures performed by eight subjects under varying lighting conditions inside a car. The dataset includes eight hundred and eighty-five intensity and depth video sequences [viv]. The dataset was recorded with the Microsoft Kinect device of resolution of 115×250 pixels and provides RGBD images.

The gesture length for each sequence in the gesture dataset is inconsistent. To create an equal length gesture, the normalization of the dataset sequence length is required. To compare with [MGKK15] the gestures were re-sampled to normal-length of thirty-two frames. If the length of a gesture sequence is less than the sequence it is reshaped into a normal-length sequence by up-sampling through repetition of frames. For sequences longer than the normal-length the gesture sequences were sub-sampled by dropping frames.

The normalized-length gesture sequences contain depth and intensity values. Intensity gradient values were calculated. The gradient and the depth values were normalized over the dataset and a two channel input from the gradient and normalized depth was created for each frame. The labels corresponding to the gesture type mark each frame. The gestures sub-sampling was done such that the the frame sequences with most variation in hand shape and motion were dropped with smaller probabilities. This is done by sampling based on magnitude of per pixel change over time within a gesture. The dense optical flow between two frames separated by time $\delta T = 2$ was calculated and the absolute change per pixel over the entire gesture was used for sampling distribution. This strategy allows improving the probability of conserving the fast changing frames during sub-sampling and increasing such frames when up-sampling the sequence.

Three classes which have 'Swiping' hand which changes direction in later parts of the gesture, and the class 'Tap three times' which is confused with the class 'Tap once' in early detection, were removed to analyze the performance. This was done because the these gestures are characteristically misidentified in short lengths. Effectively, the experiments were conducted on fifteen hand gesture classes.

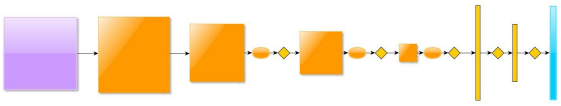


Figure 3: Single column, two input channel C3D.

2.0.1 *Short length sequences*

We focus on the improvement in latency performance for the time series so performance of classification on shorter gesture sequences was tested. To this effect, the dataset with incomplete gesture length was created. Half length and quarter length incomplete gestures were created by only using the the first sixteen and eight frames from the start of the hand motion. To assure that some hand motion indeed exists, the first two frames of the gesture sequence were always removed.

2.1 Neural Network training

All neural networks used for the experiments were trained on the negative log likelihood cost function and each uses a soft-max projection on the output layer. The networks with single column were trained for three hundred epochs and 2-column network trained simultaneously were trained for five hundred epochs. The number of epochs are chosen according to the convergence performance C3D network on the sixteen frame networks. A Negative log likelihood cost function was used for calculation of loss on each training. In case of the single phase network the training was completed in three hundred epochs. Owing to their larger size the 2-column networks are trained for five hundred epochs. One epoch is defined as the number of batches required for iteration over the full training set.

3 THE MULTI-COLUMN MODELS

3.1 The components of the Multi-column networks

The C3D layer uses volumetric convolutions. A C3D network learns 3D-filters, the two dimensions

ISBN 978-80-86943-41-1

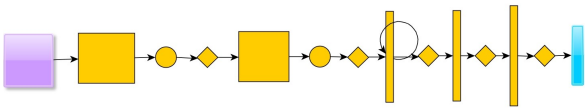


Figure 4: Single column, two input channel LSTM.

of these filters are along the dimension of the frame image and the third dimension is along time axis of the video. Accordingly, the input to the volumetric convolution layer is a block of video frames. As the convoluted blocks are propagated forward through the max-pooling layers the learnt filters reside on higher scales space. Effectively, the minimum time frame of learning in C3D is thus the time length of the spatio-temporal filter on the layers closest to the input. The LSTM on the other hand accepts sequential input. The LSTM learns to use forget gates and identifies the length of the learnt structure in the training phase.

3.1.1 C3D Network

The 2-column network of [MGKK15] uses two networks with high and low resolution input. These networks provide two sets of predictions, which are multiplied, normalized and used for a combined prediction. The C3D network used in this work is shown in Figure 3. The C3D consists of four volumetric-convolution layers, each of these layers have associated volumetric pooling layers. The tanh layers are used as the activation functions after the volumetric convolution. The fourth volumetric convolution feeds into fully connected layers which feed the outputs to the softmax layer. The softmax layer provides a probability vector as the output. The C3D provides one output for the entire block of the K stitched inputs, the output prediction in case is the index with highest probability.
For designing a network that learns to classify a gesture of length K, the input to the C3D is a $K \times 2 \times 57 \times 125$. The experiments were conducted such that each frame of the input block belonged to the same gesture type. An output probability vector of fifteen gestures is produced at the output of the C3D.

3.1.2 The LSTM Network

The LSTM network in the second column of the network has two convolutional layers followed with the usual pooling and ReLu layers. An LSTM layer and a fully connected layer follows the convolutional layers, see Figure 4. The same $K \times 2 \times 57 \times 125$ input for the C3D is feed into the LSTM. The output layer is a soft-max projection. Each frame of the gesture sequence is marked by a label such that the LSTM produces a probability output at every frame of the gesture. The LSTM predictions is made by cumulative

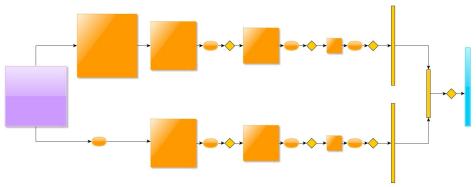


Figure 5: 2-column C3D joined at input and output.

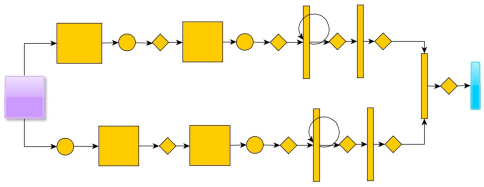


Figure 6: 2-column LSTM joined at input and output.

probability addition over the gesture sequence. The index with highest probability sum at the end of the sequence is identified as the gesture.

3.2 Experiments with Neural Network combinations

We now train , 2-column neural networks, One column of the both the C3D and LSTM networks are as described earlier in . In both cases an average pooling layer is used at the input of the second column, the remaining architecture of the second column remains similar to their corresponding first column. This is done to provide varying scales as input to the first convolutional layers of the two columns of each network. The first volumetric pooling layer in the C3D network scales only in the spatial dimensions and does not change the size of input on the time dimension.

The neural networks are trained with data from fourteen recordings from seven persons and tested on two sets of recording from the eighth person. The final accuracy results are averaged over eight experiments where all the test persons are used once. All networks are trained for full sequence(thirty-two frames) and half and quarter sequences(sixteen and eight frames)

3.2.1 2-Column Neural Networks

We performed end-to-end training with 2-column neural networks based on the components described in the Section 3.1. First, the 2-column C3D was compared against a similar size LSTM network. Thus, the networks trained for these experiments were,

- A 2-column neural network with 3D convolutional layers joined at head with a fully-connected layer, see Figure 5,

ISBN 978-80-86943-41-1

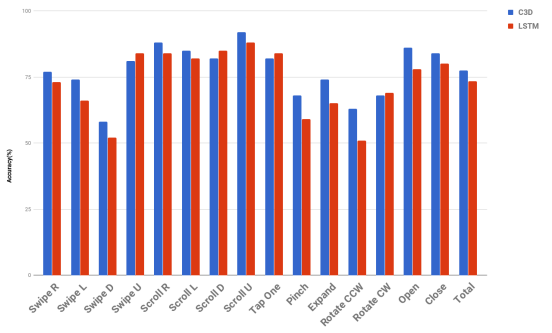


Figure 7: The accuracies of 32 Frame C3D(Red) and LSTM(Blue).

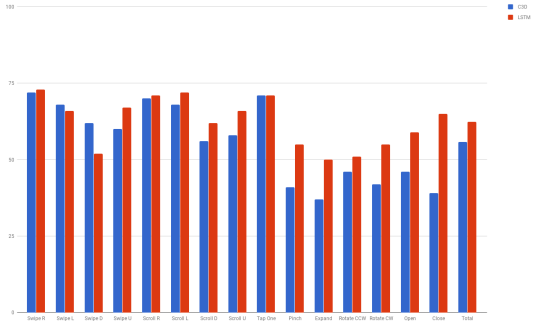


Figure 8: The accuracies of 16 frame sequence on C3D(Red) and LSTM(Blue).

- A 2-column neural network with convolutional layers followed by LSTM layer and joined at head with a fully-connected layer, Figure 6.

These networks were trained and tested for full sequence gestures of length $(K = 32)$ and half and quarter length gestures of frame length $(K = 16, 8)$.

Time Layer	Gesture Frames	Accu (%)
Conv LSTM	32	73.4
	16	**62.3**
	8	**37.3**
C3D	32	**77.4**
	16	55.7
	8	31.6

Table 1: Classification Accuracy with the 2-column LSTM and C3D.

The recorded percentage test accuracies for the two networks for the various frame lengths are reported in the Table 1. The convolutional network with the LSTM layer performed worse than the network with volumetric convolutional layers on the full sequence gestures, though the LSTM network performed better than the C3D network for shorter sequences. The results from these experiments are listed in Table 1. The class-wise classification performance of these networks on the full sequence and half length gestures is shown in the Figure 7 and Figure 8, respectively.

4 TESTING SINGLE COLUMNS

The results from the last section motivated training only the single column C3D networks and LSTM networks to identify if the behavior of volumetric convolutions and LSTM layers remain consistent. These networks were trained with the same set of inputs and labels and the initialization procedures, cost function remained the same as earlier. Apart from the two networks, another network with classical recurrent layer is also tested. The model architectures are exactly like the larger column of the neural network models described in Section 3.1. The three neural networks trained were,

- A neural network architecture from the large column of the convolutional LSTM used in 2-column experiments,
- A similar C3D network taken from the 2-column network gesture classification network,
- A neural network architecture from the large column of the convolutional LSTM used in 2-column experiments with LSTM layer replaced by a recurrent network.

The results of Table 2 demonstrate that the performance of classical recurrent neural network for the classification was poorer compared with the performance of the neural network architectures that use the LSTM layers or the volumetric convolutional layers. This is expected because an RNN network is not capable of learning long contexts.
Looking at the classification performance of Table 2, it is also apparent that the performance of the C3D reduced considerably when an early detection of gesture was made using a C3D network. The performance also deteriorated for networks with convolutional layers and an LSTM layer. An important observation is the considerably smoother decay of performance in the network with LSTM layer as compared to the C3D network. The performances of the LSTM and C3D networks on various datasets is consistent with the observations from the 2-column networks tested earlier. The C3D network performed better on the full length sequence but its performance worsens more rapidly than the LSTM network when tested on incomplete gesture sequences. Thus, both single column and the two column networks results show that the C3D performs better on full sequence gestures and the LSTM network performs better on the shorter sequences.

ISBN 978-80-86943-41-1

Time Layer	Gesture length	Accuracy (%)
Convolutional & Recurrent	32	35.6
	16	18.3
Convolutional & LSTM	32	64.6
	16	51.3
Volumetric conv (C3D)	32	73.6
	16	47.6

Table 2: Accuracy with single phase models

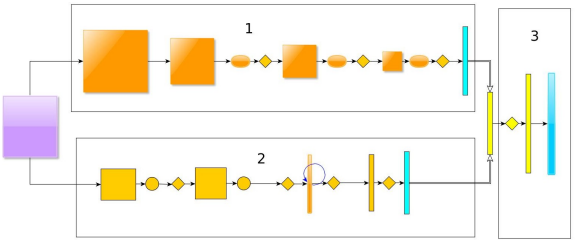


Figure 9: The proposed combination of network : Part 1 is the C3D branch; Part 2 is the LSTM branch; Part 3 is the MLP which combines output from the two temporal neural networks.

5 LSTM AND C3D COMBINATION

The observations that C3D consistently perform better on long sequence gestures, while the LSTM network always works better than C3D on shorter sequences encourages the experiments with combinations of the C3D with LSTM. The trained single phase LSTM and C3D networks were used. The output probabilities of these trained networks were combined with a separately trained MLP. The MLP learns to combine the output of the probability predictions made by the two separate networks.
The cumulative sum of the LSTM was normalized and a larger thirty dimensional vector was created by merging this resulting vector with the C3D output. The MLP is trained with an input of a thirty vector input; the output is the probability vector. The entire system is shown in Figure 9.

5.1 Training the MLP

The fifteen dimensional probability vector from the C3D and LSTM are combined together to form a thirty vector input to the MLP. The MLP has a hidden layer with sixty four nodes and an output layer of fifteen which is mapped to the softmax values. The labels of the C3D are used to train the MLP. The MLP combines the classification probability from the two networks and uses a learning rate of 0.01, is trained for two hundred epochs.

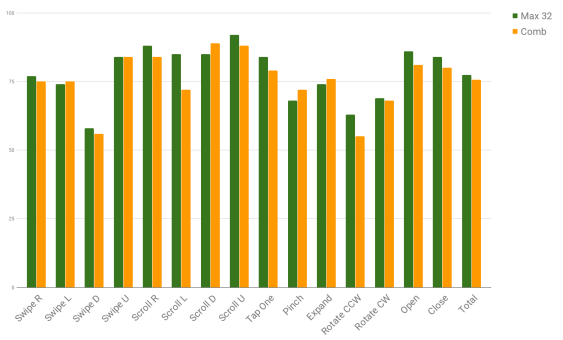


Figure 10: Class-wise average performance of 32 frame hand gestures on the Combination Network (Gold) Compared against the Best of C3D and LSTM Network (Green).

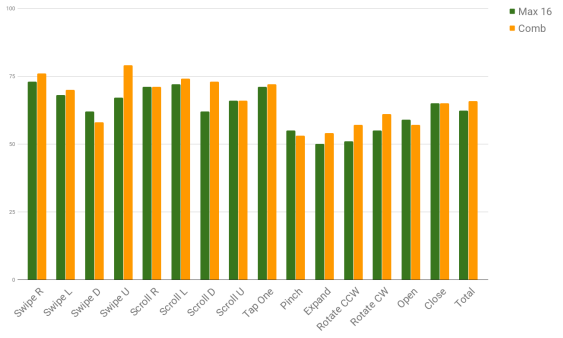


Figure 11: Class-wise average performance of 16 frame hand gestures on the Combination Network(Gold) Compared against the Best of C3D and LSTM Network(Green).

6 PERFORMANCE COMPARISON

To validate the proposed combination network, various training and test iterations were made. The networks were trained with reducing latency time. The MLP was trained separately for full length gesture of thirty-two frames, half length sequence of sixteen and for quarter length of eight frame latency. These results were compared with the best results received from either LSTM or C3D 2-column networks. The class-wise accuracies for the trained MLP network are reported in the Figure 10 and Figure 11. The accuracies are plotted for the thirty-two and sixteen frame gestures respectively. The comparisons show that the performance of the combination network is better than either of the two networks in most classes in the incomplete gesture identification problem. It was observed that the combination network had better accuracy than the best of LSTM and C3D in seven of the fifteen classes when the experiments were conducted on the full length gestures. On the other hand, when the experiments were conducted on the half gesture length starting from

ISBN 978-80-86943-41-1

Gesture length	Combination NN Accuracy(%)	LSTM (%)	C3D (%)
32	75.6	73.4	**77.4**
16	**65.7**	62.3	55.7
8	**39**	37.3	31.6

Table 3: Classification with the combination of C3D and LSTM compared with LSTM and C3D; the accuracy of best network is bold.

the beginning the performance of the combination network was better than the best of the two networks on twelve of the fifteen classes.
The average accuracies achieved in the the experiments conducted on the full, half and quarter gesture lengths are reported in the Table 3. These values are compared against the performances of the 2-column LSTM and C3D tested in the Section 3.

The results of this combinatorial networks tabulated in Table 3 demonstrate that the network performs slightly worse than the two column C3D network in case of long gestures. However, the combination network outperformed the two-column LSTM based gesture classifier in every scenario. When classification accuracies were evaluated at shorter latency period it was observed that the combinational network performed better than the 2-column C3D network. For a half length gesture sequence the accuracy of the combinatorial network was 10% higher than the C3D network (reported in Table 3), it was also marginally better than the LSTM network by 3%.
The combination of the block learning property of the C3D with the contextual learning of the LSTM network may explain the improved performance of the network on shorter incomplete sequences. The accuracy results for the experiments conducted on the one-fourth length sequences demonstrated similar results. The results of the quarter gesture dataset also demonstrated the difficulty of early identification of the gestures. It is clear that the accuracy rates falls dramatically as the sequence length is reduced.

7 DISCUSSION AND CONCLUSION

When the models are tested in the forward phase on a CPU the proposed network with sequential input to the MLP does not return a real time performance. A smaller model means less computation cost in a system embedded in the automobile. More importantly, we wished to understand the generalization behavior on reducing the size of a 2-column neural network for gesture recognition. So, apart from the large combinational model, we trained a smaller model on the same dataset. It included two volumetric convolution layers and two linear layers apart from the output log softmax layer in one branch, and one volumetric-convolutional layer, and a fully connected layer in the other branch. It was identified that choice of the initial learning parameters for a smaller network is crucial. The performance of such a network was generally worse.
We tested this network for 32 frame and 16 frame gesture classification problem. It was recognized that that a combinational network with smaller contributing networks performs considerably worse than the larger network.

Gesture length	Accuracy(%)
32	53
16	42

Table 4: Classification with the combination of smaller C3D and LSTM Networks.

7.1 Conclusion

This work showed a possible method for improving fast identification of hand-gestures. It proposed a possible combination of the C3D and an LSTM network. We show an improvement in the early classification performance. The proposed combinational network performs better as compared to existing state-of-art C3D neural networks by over 10% when applied for early identification of hand gesture sequences. It is shown that the C3D network performs better than LSTM on fixed length full gesture sequence, but LSTM performs better than the C3D network on incomplete sequences.
We demonstrated that a combination of such sequential learning and time filtering networks can improve the classification performance on shorter sequences.
The model for the combination of C3D and LSTM can be extended further and the proposed example should encourage further investigations. This work uses discontinuous windows while training and testing the model. This choice is constraint to a fixed input size. It is possible to use a sliding window approach for sampling while training and testing. Such an approach would allow working with gestures of variable sizes. A system of this nature should have the capacity to handle unsegmented gestures.

8 ACKNOWLEDGMENT

This work is supported by the National Research Fund, Luxembourg, under the AFR project 7019190.

9 REFERENCES

[AAGES10] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny.

 ISBN 978-80-86943-41-1

An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[CFH03] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and vision computing*, 21(8):745–758, 2003.

[CLS15] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3218–3226, 2015.

[CMS12] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[DAHG+15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[GYB04] S Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 35–35. IEEE, 2004.

[KKKA13] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[LGS08] Yun Liu, Zhijie Gan, and Yu Sun. Static hand gesture recognition and its application based on support vector machines. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on*, pages 517–521, Aug 2008.

[MGKK15] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2015.

[MGKP15] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Kari Pulli. Multi-sensor system for driver's hand-gesture recognition. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.

[OBT14] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2368–2377, 2014.

[RSP11] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2011.

[TTGS16] Aditya Tewari, Bertram Taetz, Frederic Grandidier, and Didier Stricker. Two phase classification for early hand gesture recognition in 3d top view data. Springer, 2016.

[viv] Viva. http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-gestures/.

[WKSL13] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[YH15] Jiachen Yang and Ryota Horie. An improved computer interface comprising a recurrent neural network and a natural user interface. *Procedia Computer Science*, 60:1386–1395, 2015.

 ISBN 978-80-86943-41-1

Finding Similar Movies: Dataset, Tools, and Methods

Hongkun Leng[1], Caleb De La Cruz Paulino[1], Momina Haider[1], Rui Lu[1],
Zhehui Zhou[1], Ole Mengshoel[1], Per-Erik Brodin[2], Julien Forgeat[2], Alvin Jude[2] ‡

Carnegie Mellon University[1]
Silicon Valley
U.S.A.
{hongkunl,cdelacru,mominah,rlu1,
zhehuiz}@andrew.cmu.edu,
ole.mengshoel@sv.cmu.edu

Ericsson Research[2]
Silicon Valley
U.S.A
{per-erik.brodin,julien.forgeat,
alvin.jude.hari.haran}@ericsson.com
(‡ corresponding author)

ABSTRACT

Recommender systems are becoming ubiquitous in online commerce as well as in video-on-demand (VOD) and music streaming services. A popular form of giving recommendations is to base them on a currently selected product (or items), and provide "More Like This," "Items Similar to This," or "People Who Bought This also Bought" functionality. These recommendations are based on similarity computations, also known as item-item similarity computations. Such computations are typically implemented by heuristic algorithms, which may not match the perceived item-item similarity of users. In contrast, we study in this paper a data-driven approach to similarity for movies using labels crowdsourced from a previous work. Specifically, we develop four similarity methods and investigate how user-contributed labels can be used to improve similarity computations to better match user perceptions in movie recommendations. These four methods were tested against the best known method with a user experiment ($n = 114$) using the MovieLens 20M dataset. Our experiment showed that all our supervised methods beat the unsupervised benchmark and the differences were both statistically and practically significant. This paper's main contributions include user evaluation of similarity methods for movies, user-contributed labels indicating movie similarities, and code for the annotation tool which can be found at http://MovieSim.org.

Keywords

Recommender Systems, Item-Item Similarity, Crowdsourcing, Supervised Learning, MovieLens.

1 INTRODUCTION

The Role of YML and MLT Recommender Systems. With the increase of online retail stores with massive offerings, users can easily get lost and suffer from information overload. Recent advances in machine learning have provided methods to assist users in these extremely large online stores. This is typically done by reducing their visible size to what is cognitively manageable by the users, by only surfacing the items most relevant to them. The most common approach to do this is with Recommender Systems (RSs). The idea behind RSs is to use past user interaction data to predict what they will want or like, and only present (or display) those items. Netflix, for example, has been quite vocal about their use of RS techniques, and have claimed that they improve user experience in general by allowing users to quickly find movies they want to watch [BL07].

We can partition RSs based on two features they can potentially provide: "You May Like" (YML) and "More Like This" (MLT). In a YML RS, users are shown a list of items they are predicted to enjoy, based on preferences they have provided for other items. MLT recommendations, on the other hand, generally surface when a user selects one specific item. Here they are usually presented with a list of details about the item; for a movie, details would include the name, description, director, year it was released, and so on. Below such details there is generally a list of other similar items, with a header such as "More Like This," "Similar to This," "You May also Like," "People Who Liked This also Liked," and so on. This presentation is analogous to walking into a physical store, going right to the item you want to purchase, and then being able to look at other similar items nearby to aid in the decision making process.

YML RSs generally rely on user consumption data in order to build machine learning (ML) models. This research area gained popularity after Netflix released a massive amount of user-to-item ratings data [BL07].

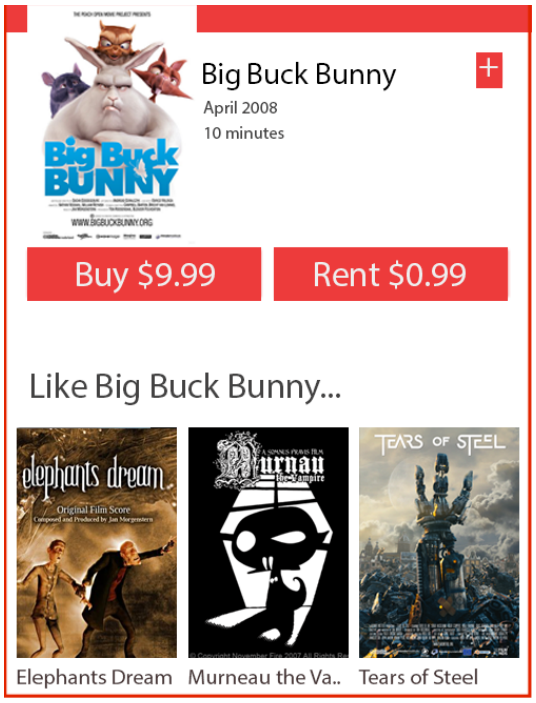


Figure 1: An example of similar movies used to perform "More Like This" recommendations. Image from [CDL16], used with permission.

After this movie dataset was retracted for legal reasons, the RS community built a replacement dataset called MovieLens [HK16]. Similar datasets have been released for other domains including music [JDE07; DWE05]. These datasets have made it possible for ML and HCI researchers to study the impact of different RS algorithms on user experience. The MovieLens website[1] even allows anyone to create an account and submit movie ratings as a way to contribute to RS research and development.

MLT has, compared to YML, received less attention. In the cases where researchers have required a means to find similar movies, they have often defined similarity heuristically based on metadata instead of taking a data-driven approach. For example, researchers have defined that two movies are similar if they share *genre*, *cast*, and *director* [BHY97].

The Need for User-Centric MLT Recommenders. We we see a major difference between how YML and MLT RSs are built up to this point. A YML RS often uses a data-driven approach while an MLT RS is often defined by the designers. This approach goes against general HCI principles including the slogan that "designers are not users." Clearly, there is an opportunity to bring in more data into MLT RSs [Nie08]. Since user interactions are increasingly shaped by ML methods and models, HCI and ML researchers need to work together to ensure that the ML methods that interface with users are evaluated and improved with the user in the centre. We believe there are two plausible reasons for the lack of focus on MLT despite its pervasive use in current-day technology: (1) There is a lack of movie-movie similarity datasets, which hinders work by ML researchers. (2) There is also a lack of validation that datasets or labels could even be useful to improve the user experience. Taken together, (1) and (2) form a vicious cycle, since we need datasets to improve user experience, and it is hard justifying a large-scale data collection activity without knowing if labels could even be helpful to users.

Our Contributions to MLT Recommenders. We hope, via this work, to break this vicious MLT cycle by collecting and releasing similarity data for movies, and by showing that data-based similarity predictions can match users' perception of similarity. Datasets and tools can be found at the project website [2] or by contacting the corresponding author. We believe this is the first paper to demonstrate the benefit of this data-driven approach to movie similarity.

Our hypothesis was that the current methods used to find similar movies can be improved if we use a data-driven approach, where labelled data is used to build supervised machine learning models. These supervised methods built on user-contributed data indicating perceived similarity would better match users definition of similarity, lead to improved perceived similarity and therefore improved experience.

In this paper, we use a small dataset from Colucci et al. [CDL16] to learn four different ML models. Experimentally, we show that the four different models can be used to predict movie similarity in a way that is consistent with users' perceived similarity. Specifically, our main contributions are: (1) Empirical evidence that ML models can be used to predict similar movies in a way that more closely matches user perception than previous work. (2) An evaluation of four ML methods that can be used to build lists of similar movies. (3) A novel dataset containing 6605 binary labels and intermediary data required to build a list of similar items for movies in MovieLens 20M.

The Structure of this Paper. We discuss related work in Section 2. In Section 3 we present machine learning methods used to learn similarity models from data. The design of the user test to evaluate those similarity models is laid out in Section 4, while the results of the study are presented in Section 5. The paper is wrapped up with discussions in Section 6 and conclusions in Section 7.

[1] https://movielens.org

[2] http://MovieSim.org

ISBN 978-80-86943-41-1

2 RELATED WORK

The first question we asked was "what makes people believe two items are *similar*?" Recent advancement in psychology and cognitive science support the notion that people use a dual-process model, whereby perceptions of similarity is built on a combination of feature-based taxonomic relations, and relationship-based thematic relations [WB99]. Taxonomic or hierarchical relations are based on internal characteristics, such as features of the items themselves, while thematic relations are external; there is a separate event or scene that connects the two items. For example, cars and motorcycles are taxonomically similar since they share many features; both have engines, wheels, and fall under the category of "ground transportation". Motorcycles and helmets are thematically similar since they are often used during the same event, i.e. a person riding a motorcycle [EGG12]. Individuals appear to favour either thematic or taxonomic similarity, and at varying levels, and with an individual's preference remaining the same even across different concepts [MG12].

Similarity algorithms (or "methods") are generally built on the intuition that "two objects are similar if they are referenced by similar objects" [JW02]. Two common methods are item-item collaborative filtering (I-I CF) and content-based (CB) similarity. In I-I CF, items are considered similar based on their relationship to users. E.g. two movies would be considered similar if they are both watched or similarly rated by a similar group of users [MMN02]. Although the term similarity is often used in item-item CF, it was originally developed to recommend items to users i.e. YML recommendations [SKK01] and not MLT. Researchers often pre-generate a lists of similar items built with CF similarity to perform YML recommendations [Kar01; SKK01; CZG16]. This has been shown to be 27% better and 28× faster than the traditional user-neighbourhood based RS [Kar01]. In the CB approach, items are considered similar if they possess similar attributes [CZC15]. For movies, these usually comprise of *genre*, *director* or *cast* [PJH14; SPU02]. CB similarity could alternately be done with tags or keywords, contributed by users or domain experts; MovieLens released a set of user-contributed tags for movies via the Tag Genome Project [VSR12]. CB-similarity can also perform YML recommendations [CZG16; NK11], and improving CB similarity using supervised learning can improve YML recommendations [WAL17]. Both approaches have its own downsides; CF requires user data making it unsuitable for new items, while CB could produce only obvious recommendations. CF-CB hybrids could potentially overcome these limitations [DDV14].

Human judgement can be used to assess similarity. An absolutely correct ground truth is unlikely since the notion of similarity is subjective, but researchers aim to reach a consensus or a 'generally agreeable classification' [OD15]. Human judgement has been used to label similarity in music [JDE07; DWE05], birds [WBM10] and geometric shapes [JLC09] among others, usually to find similarity methods that match user perception. Given the massive amount of labels required for this, researchers have also investigated how to elicit confident labels at a cost-effective manner [WKB14]. We have found very few works related to movie similarity from a users perspective. In one study researchers had participants rate the similarity between 910 movie posters, but this task was for image similarity rather than movie similarity [KGA16]. In another study, researchers used low-level features in the form of subtitles to find similarity between movies [BG16].

Experiments designed to evaluate similarity in both computer and cognitive science often elicit labels in one of two ways: relative or pairwise. In relative similarity, raters are asked "is X more similar to A or B". While in pairwise similarity, raters are asked how similar is the pair X and A and then separately how similar is the pair X and B [McF12; FGM15]. Pairwise assessments often use binary labels or Likert-like scales. Researchers in music similarity found that items received more consistent labels when two levels were used ("Similar", "Not Similar"). However, the participants were more consistent with their labelling when three levels were used ("Very Similar", "Somewhat Similar", "Not Similar") [JDE07]. A binary scale is seen as less complex [DGL11] and took less time to complete [GNZ07] without compromising quality.

Research in similarity has benefited by borrowing experimental design and evaluation metrics from the Interactive Information Retrieval (IIR) community which prioritises the user in IR tasks [Kel09]. We consider IIR and item-item similarity to be analogous; in both cases the user performs a query, and receives results relevant to that topic, usually in an list ranked by estimated usefulness [Sin01]. With MLT, the query and the results are the same type of object, making it comparable to the Query-By-Example (QBE) approach [Tre00]. A comprehensive study on similarity, MLT and QBE as it relates to music can be found in [McF12], which demonstrated how elicitation of labels can improve similarity and recommendations in music.

Clough and Sanderson present a comprehensive overview of the many ways in which IR systems can be evaluated [CS13], one such method is Mean Average Precision (MAP) [SAC07]. Precision itself can be measured as either of the following [MRS08, Chapter 8]:

$$\frac{(\text{number of relevant results})}{(\text{number of results})} \tag{1}$$

 ISBN 978-80-86943-41-1

or

$$\frac{(\#\text{true positives})}{(\#\text{true positives} + \#\text{false positives})} \quad (2)$$

The difference in the two is in the denominator: Equation 1 includes all items returned, regardless whether or not they were labelled. We used Equation 2 which only includes items explicitly labelled true or false. Mean Average Precision first evaluates the precision of a topic (or in our case a movie), and then calculates the mean over all topics. IIR systems can be evaluated from a system perspective, which measures how well the system can rank items, or from a user perspective which measures the user satisfaction with the system [Voo01]. It has been argued that MAP is a system metric since it evaluates performance based on topics, while a more suitable measure for user satisfaction involves assessment of relevance of a fixed number of k items, such as precision@k [MRS08, Chapter 8]. However the D&M Information Systems success model introduced in 1992 [DM92] and revisited 10 years later with a survey of almost 300 journal articles [DM03] demonstrated that information quality –including relevance– leads to better user satisfaction. Thus we ourselves see MAP as a direct measurement of system performance and an indirect measurement of user satisfaction. In our research, we fix the number of items produced and evaluated per method, hence effectively measuring MAP@k which makes it a more suitable measure for user satisfaction as per recommendations above.

3 BUILDING SIMILARITY METHODS

Our goal was to compare supervised similarity methods against unsupervised methods with a user test. Here we first describe the previous work [CDL16] for context as it supplied the labels used, inspired the user interface of our study, and provided a benchmark against which we would compare our methods. Then we how we built and tested methods offline to decide which machine learning methods and features should be used in the user study, which is presented in Section 4 and Section 5.

3.1 Existing Dataset & Methods

Colucci et. al [CDL16] evaluated existing movie similarity methods from a user perspective, and showed these methods matched user perspective about half the time. They implemented four similarity methods, two based on CB similarity and two based on CF similarity. The two CF approaches were based on works by Sarwar et. al [SKK01], and used user contributed ratings of movies in MovieLens. One CF method used Pearson's correlation and another method used cosine similarity; both were implemented via the LensKit libraries from MovieLens[ELK11]. We will refer to these methods as CF-Pearson and CF-cosine respectively. The authors also implemented two CB similarity methods. The first was a blackbox from TheMovieDatabase (TMDb),[3] which used a combination of *genre* and user-contributed keyword, built with Solr's MoreLikeThis feature. The second CB approach used movie metadata from the Open Movie Database (OMDb)[4] as input, with similarity calculated using TF-IDF. Each column was weighted as follows: *title* 0.25, *genres* 0.2, *cast* 0.2, *writer* 0.15, *director* 0.1, and *plot* 0.1. We will refer to these methods as CB-keywords and CB-metadata respectively. The authors exposed a web-based front-end where participants could label the movies "similar," "not similar," or to skip if they didn't know. The results of each methods are shown in Figure 2. CB-keywords was the clear winner in their research, although we note that the authors used pairwise precision and not MAP.

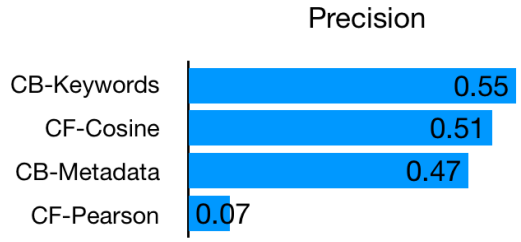


Figure 2: Perceived similarity of existing methods reported by [CDL16].

The primary purpose of their research was to evaluate similarity methods, but a byproduct was labels indicating perceived similarity. There were specifically 3803 binary labels from 14 graduate students, which we would later use to train our methods. We took a few cues from this work, with a goal of improving on it. First we used the labels collected to build and evaluate supervised machine learning models, which we hypothesised will perform better than their unsupervised methods. Second, we reused CB-keywords as-is as the benchmark in our study with the goal to outperform it. Third, we built the web front-end shown in Figure 3 to mimic the previous study.

3.2 Our Methods

Here we describe how we built and evaluated our supervised learning methods offline, with the goal of selecting the best ones for inclusion in the user study. Learning-to-Rank methods were used to produce a list of similar items where more relevant items are higher on the list [QLX10]. We tested permutations of methods described below, and selected the best methods and features combinations for the user study. All evaluation was done with leave-one-out cross validation, where

[3] https://www.themoviedb.org
[4] https://omdbapi.com

ISBN 978-80-86943-41-1

one movie (not one label) was left out, since the goal was to optimise MAP and not pairwise precision. Since there were 143 movies in the dataset, the results presented below are those averaged over 143 iterations.

We wanted to focus on CB similarity, but also aimed to build a hybrid model where similarity is predicted based on a combination of CB and CF similarity. Like Colucci et. al, we used metadata from OMDb and started with the same features: *title*, *genre*, *cast*, *writer*, *director*, and *plot*. We then added these features also from OMDb: *awards*, *country*, *full plot*, and *language*. The rationale is that two movies could be considered more similar because they were both in Mandarin, both from France, or both won the Independent Spirit Award. The *full plot* was longer than *plot* and therefore could have more relevant keywords. Previous work [CDL16] proposed that an older candidate movie may be seen as less similar than a newer one, so we engineered the feature *age difference*. Let M_1 and M_2 be the two movies for which we are evaluating similarity, then:

$$\text{Age Diff} = 1 - \frac{|\text{releaseYear}(M_1) - \text{releaseYear}(M_2)|}{\max(\text{ageDiff})} \tag{3}$$

max(ageDiff) refers to the difference between the latest M_1 and the oldest M_2 in the entire database.

We know that movie pairs with high CF similarity can be perceived to be similar [CDL16]. We further believed that CF and CB can be hybridised to produce a single method that considers both, where two movies are considered similar if they shared metadata *and* had common raters. This could reduce the possibility of an actually similar movie excluded by CB due to limited overlap in the metadata, or by CF because it has too few ratings (e.g. new movies). Of course movies that simultaneously suffer from both issues cannot be addressed by this hybrid approach. We considered using LensKit for CF, but there we observed one major issue we could not solve. Since CF-Pearson and CF-cosine used individual ratings of a movie as input, it did not work well when two movies have too few common raters. We believe this explains why CF-Pearson performed poorly before [CDL16]. Possible solutions such as adjusting the formulation or including a regularisation term was outside our scope. We instead shifted to using matrix factorisation (MF) methods, which adjusts for the number of ratings using latent factors. We tried two libraries which performed MF for RSs: myMediaLite (MML) [GRF11] and libMF [CYY16]. Both have their own benefits: MML was built specifically for movies while libMF was built for speed.

Three approaches were considered to calculate similarity between each features (e.g. *genre*): TF-IDF, BM25F and Jaccard similarity. While TF-IDF is a common approach, BM25F is a reasonable alternative. Three different supervised learning methods were considered for supervised learning methods: linear regression, logistic regression, and SVM with linear kernel. We wanted to try all candidate methods with CF included as a feature and without it. Since there were two CF similarity libraries to consider, we had in fact three factors: none, libMF and MML.

3.3 Method Selection

Now we move on to selecting the best methods among those described in Section 3.2 to be included in the user testing phase of our work. Note that we have considered three similarity approaches (TF-IDF, BM25F, Jaccard), three supervised learning techniques (linear, logistic, SVM) and three ways to build CF similarity (none, libMF, MML). Trying all methods would have required us to evaluate ($3 \times 3 \times 3 = 27$) methods offline, which was too computationally expensive. So we first aimed to eliminate the least performing methods.

Between the three similarity approaches, TF-IDF provided the highest MAP on average at 0.73 followed by BM25F at 0.72 and Jaccard at 0.69. In addition to having the lowest MAP, Jaccard was also unusually slow, so it was eliminated. The three supervised methods were virtually indistinguishable; linear regression had an average MAP of 0.73, logistic regression 0.74 and SVM 0.73. We decided to only use linear regression as it was easier to explain and closer in implementation to previous work. We now had two similarity measures: BM25F and TF-IDF, and one supervised learning method: linear regression. This brought it down to a more manageable ($2 \times 1 \times 3 = 6$) methods.

We found that the top three combinations by MAP were TF-IDF + no CF (0.71), BM25F + no CF (0.71) and BM25F+MML (0.70), and decided that these would be included in the user testing. There was little difference noticed when CF was included as a feature or not. But there were differences in compute time for different MF libraries: libMF took 1.3 minutes while MML took 11.4 minutes on average per iteration. We decided to include BM25F+libMF (0.65) in the user testing because we believed libMF had its benefits. Firstly libMF was almost $10\times$ faster than MML; second BM25F+libMF produced a very different list than other methods, with a Jaccard Difference to BM25F of 0.48. For context BM25F+MML had a Jaccard Difference of 0.07 to BM25F. So, while BM25F was slightly less precise, it did provide quite a different list. Since diversity is known to improve user experience with YML [ZMK05; VBK14], we thought perhaps this diversity by libMF could benefit MLT too.

4 USER EXPERIMENT DESIGN

The goal of our user testing was to validate if our supervised methods could lead to better perceived similarity. The four methods chosen, as discussed in Section 3, are

 ISBN 978-80-86943-41-1

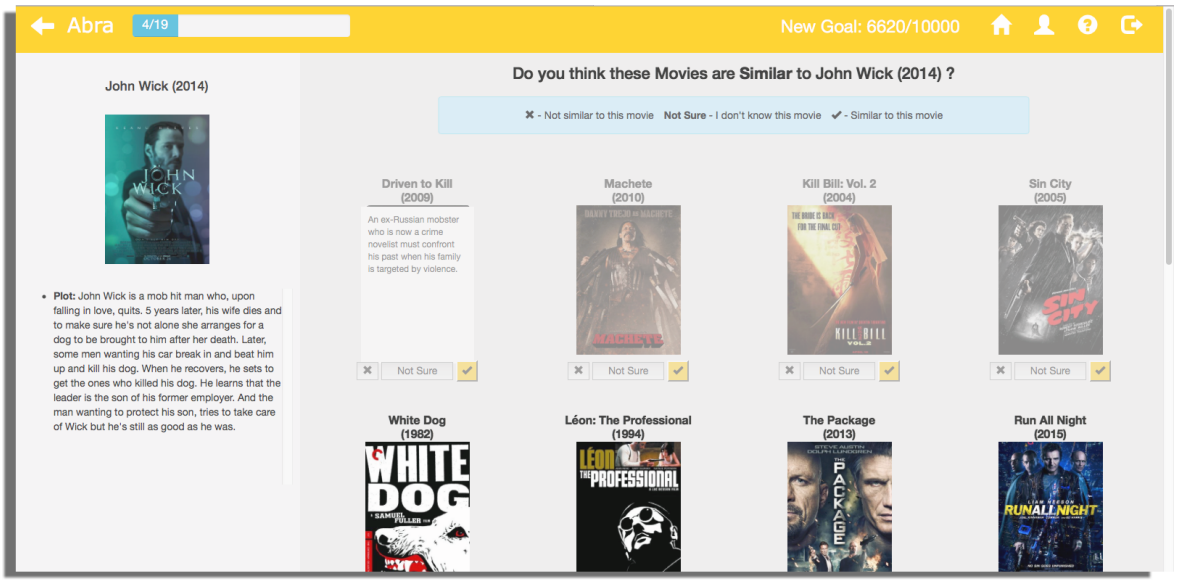


Figure 3: Experiment's user interface as per Section 4. To the left is the poster and short plot of the selected movie. To the right are eight suggested similar movies and hovering over a poster will surface a short plot. Users submit relevance feedback by indicating if the suggested movie is similar to the movie to the left. Labelled movies are greyed out. In this example there are 19 suggested similar movies, and four have been labelled.

TF-IDF, BM25F, BM25F+MML, and BML25F+libMF. We also included the best method from previous research as a benchmark [CDL16] (See section 3.1 and Figure 2). Our user experiment therefore simultaneously assessed five similarity methods. We built a publicly accessible website where anyone can sign up and submit annotations. Like the previous work [CDL16] the database of movies contained about 27000 movies from MovieLens with metadata from OMDb. We collected no personally identifiable information, except for some optional demographic information including work, age, and gender to evaluate diversity. We spread news of the website via social media to elicit volunteers.

After users signed up and completed demographic information, they were taken to the landing page. A randomly generated list of movies were shown as a suggestion. The user's ID was used as a seed to the randomiser to ensure the list does not look random at every refresh. A Bayesian belief network was used to ensure movies shown were representative based on genre, popularity and age. A search bar allowed users to find any movie by title. The page also showed a "goal" indicating the number of labels we wanted, and the number currently available. The goal was set to 5000, and increased in increments of 5000 when each stage was 80% reached.

Upon selecting a movie to evaluate, a user was taken to the annotation page shown in Figure 3. Six similar movies were chosen per method and merged into a list to remove duplicates. This merged list was randomised before being shown to users. Note that in the extreme case where all methods produced the same list, only six movies would be shown. In the other extreme case where all methods produced a different list, 30 movies would be shown. The number six was selected as it produced a total of 20 similar movies in the merged list on average, which is in line with the number of similar movies surfaced in previous work [CDL16].

Users were requested to supply labels by indicating if the suggested movies were similar. They could indicate the movie was similar, not similar or "not sure". Users were able to undo any actions or change any labels at any time. Candidate similar movies displayed the title and year, the plots were available on-demand. Plots were shown when they hovered over the poster with a mouse or touched the poster on a touchscreen.

5 USER EXPERIMENT RESULTS

Here we discuss the results of the user testing according to the design in Section 4. We start by describing the responses, then we evaluate the performance of similarity methods, and finally present an analysis.

5.1 Responses

A total of 136 people signed up and 114 people participated in the survey indicating a drop-out rate of 16%. Exactly 100 participants reported age, both the median and mode was 24. 72 reported their gender as male (63%), 30 as female (26%), 1 reported "others" and 11 chose to not report gender. The participants had a high education rate, with 69 in or completed a graduate program, 25 bachelors, 9 high school, and 11 not reported. In terms of employment, 65 self-reported as students while 29 were employed, 7 unemployed, 1 each reported "retired," "self-employed," or "homemaker."

ISBN 978-80-86943-41-1

	Full			In training set			Not in training set		
Method	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**
TF-IDF	0.70	0.32	0.80	0.75	0.27	0.80	0.69	0.33	0.80
BM25F	0.69	0.32	0.78	0.72	0.30	0.79	0.68	0.33	0.76
BM25F + MML	0.70	0.32	0.80	0.73	0.30	0.82	0.70	0.32	0.80
BM25F + libMF	0.65	0.36	0.71	0.75	0.26	0.76	0.62	0.36	0.67
Benchmark	0.48	0.35	0.50	0.47	0.34	0.40	0.49	0.36	0.50

Table 1: Mean Average Precision (MAP) along with standard deviation (SD) and median of each method. The first group shows the full results, the second group represents those movies that were strictly in the training set while the final group excludes all movies that were in the training set.

Participants submitting a total of 9511 responses for 393 movies, of which 6605 were binary (yes/no) while 2906 were 'not sure'. 310 of the movies were not in the training set, allowing us better claims of generalisability. On average, each user contributed a mean of 83 labels. The search function was used a total of 128 times by 28 users. There were 1087 movie pairs with binary responses from more than one user. We checked for agreement and found that 702 or 65% had complete agreement, i.e. everyone who labelled these pairs agreed that the pair was similar (or dissimilar). while 848 or 78% had at least a $2/3$ agreement.

We analysed our responses to see if it is representative of the MovieLens dataset in terms of *genre* and visualised in Figure 4. From this image we see that the movies labelled in our study is at least more representative than Colucci et. al; the top two *genre*s are almost identical. A Pearson's correlation with *genre* percentage as input showed that our dataset had $r = .96$ against all movies from MovieLens, and $r = .90$ against movies from the past 5 years of MovieLens. In contrast Colucci et. al had a correlation of $r = .49$ and $r = .55$ respectively. We consider this to mean our study is more representative in terms of *genre*.

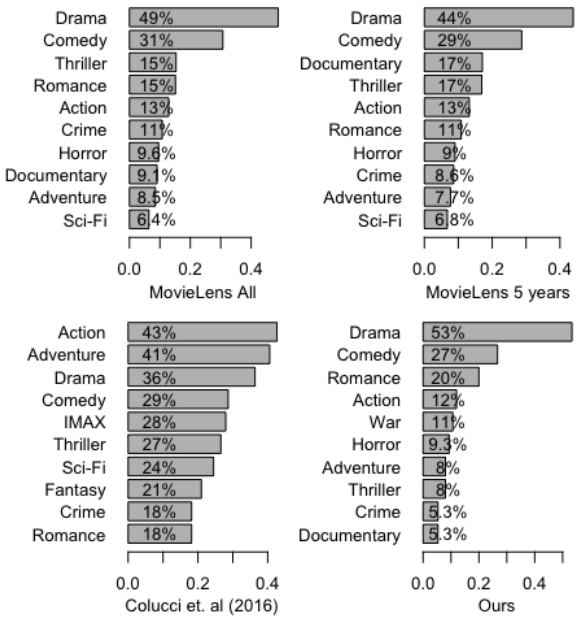


Figure 4: Top: Percentage of movies containing these *genre*s in the entire MovieLens library (L) and movies from the last five years of MovieLens (R). Bottom: Distribution for Colucci et. al [CDL16] (L), and ours (R).

5.2 Performance

We analysed the results by three groups: the first contained all movies, the second group were only where the selected movies were part of the training, while the third group were those where the selected movies were not in the training set. The last group was most important as it indicated generalisability. We used a Kruskal-Wallis test for statistical significance as our experiment used ordinal responses. This test is based on median, which we therefore report alongside the means.

We see in Table 1 that all four methods introduced in this paper outperformed the benchmark in all three groups. We performed a Kruskal-Wallis test to check for statistical significance and found that the difference was statistically significant for all three groups full ($\chi^2 = 86.868, df = 4, p < .001$), in training set ($\chi^2 = 35.429, df = 4, p < .001$) and not in training set ($\chi^2 = 58.016, df = 4, p < .001$). Hence we ran a post-hoc test with Dunn's t-test and Holm-Bonferroni correction. All pairwise evaluations involving the benchmark were statistically significant ($p < .05$) while those that did not involve benchmark were not. Practical significance between the benchmark and our methods in Cohen's d, is in Table 2. A common interpretation of Cohen's d is that .2 means the practical significance is small, .5 is medium but visible to the naked eye, while .8 is considered large [SF12].

	Full	In train.	Not in train.
TF-IDF	0.63	0.90	0.57
BM25F	0.61	0.79	0.57
BM25F + MML	0.65	0.81	0.61
BM25F + libMF	0.47	0.92	0.36

Table 2: Effect size in Cohen's d against the benchmark. Grouped by all movies, movies strictly in the training set, and movies strictly not in the training set.

ISBN 978-80-86943-41-1

5.3 Analysis & Implication

The main findings is that the use of labels to train a supervised model results in an improvement in perceived similarity. In our case, even a small training set from a few users was significantly better than an unsupervised model.

An interesting finding to us was that there were no statistical significance between different models. The biggest difference was between BM25F+libMF vs. BM25F+MML for items not in the training set, with a difference in MAP of 0.08. It had a Dunn's post-hoc test of $p = .2016$ and effect size measured with Cohen's d of .22. We believe that this comparison could be statistically significant with a small effect size in a live deployment or another experiment with larger number of movies and participants

Movies that were in the training set appear to have higher precision and performed better against the benchmark in Cohen's *d*. This is unsurprising but highlights the importance of evaluating such methods based on items not in the training set to infer generalisability.

6 DISCUSSION

Our experiment shows user contributed labels are useful in building similarity models. We believe there can be reasonable confidence in the experiment presented here. There was a high number of movies in our evaluation that were not in the training set, which speaks to the generalisability of the methods. The inter-rater agreement of 65% for complete agreement indicated reliability of participants. The distribution of movies by *genre* were also representative of movies in the library.

The fact that BM25F+libMF had lower MAP than other supervised methods was unsurprising considering it had slightly lower performance during the machine-learning stage. We note that this method showed the highest difference when comparing movies in the training set and those without, which points to overfitting and reinforces the need to test methods against users rather than stop after the machine learning phase. We previously pointed out that this method is faster than MML and produced a notably different list of similar movies. It could therefore still be useful if timeliness and resource consumption matters, or by researchers eager to test out different variations quickly. It could also be used as part of an ensemble to produce a more diverse list of similar items.

We were admittedly surprised that BM25F showed no improvement over TF-IDF. In fact both TF-IDF and BM25F had the exact effect size over the benchmark measured in Cohen's *d*. There was also no noticeable difference in the time it took for both to complete. The inclusion of collaborative similarity as a feature in the hybrid method BM25F+MML seems to have some improvements noted in the effect size for movies not in the training set. But this was not statistically significant and admittedly lower than we expected. Future researchers could investigate this further, including to identify which types of movie benefit from the hybrid approach. For now, our recommendation to researchers using our generated list of similar item should use BM25F+MML, while researchers who wish to build from scratch using our similarity labels only could start by implementing TF-IDF.

This work opens up a number of research question which we encourage future researchers to explore, we list a few such questions here. We believe more work needs to be done to understand *why* people believe two movies are similar. This could be used to build better machine learning methods including personalised similarity methods and to provide a better experience overall.s It may be possible to analyse our data to identify which features are most salient in predicting similarity, and likewise if different people have different preferences. We believe a similar study could be done to improve similarity in TV series, songs, or video games.

It is evident here that labels are useful in the prediction of similarity. While this is not surprising, this is the first time it has been shown to be true in finding similar movies. Future work should include elicitation of labels from many more subjects and for many more movies to ensure higher coverage and better confidence. We therefore release all labels collected during this study, source code for the website to elicit labels, and all intermediate data generated during the process in order to encourage other researchers to build on our work. Our dataset is a notable improvement over Colucci et. al [CDL16] in terms of number of labels, number of participants, and genre representation. We hope this would lead to even better machine learning models, which will improve user experience by helping them find similar movies.

7 CONCLUSION

In this paper we showed that finding similar movies can be improved if we use human-annotated data representing perceived similarity in movies. We tested a few machine learning options offline, identified the best methods and features, and then evaluated with users. Our methods demonstrated significant improvement over the benchmark introduced in previous work which was built with unsupervised machine learning. The four supervised methods in our user testing were not statistically significant between themselves, which indicated that in a small sample such as ours any of our methods could produce the same experience. We showed that there is more that can and should be done to improve user experience with similar items. We release our dataset to encourage and enable future research in this domain by both HCI and ML researchers.

 ISBN 978-80-86943-41-1

REFERENCES

[BG16] Bougiatiotis, K. and Giannakopoulos, T. Content Representation and Similarity of Movies based on Topic Extraction from Subtitles. Proceedings of the 9th Hellenic Conference on Artificial Intelligence. ACM. 2016,

[BHY97] Burke, R. D., Hammond, K. J., and Yound, B. The FindMe approach to assisted browsing. IEEE Expert 12.4 (1997),

[BL07] Bennett, J. and Lanning, S. The netflix prize. Proceedings of KDD cup and workshop. Vol. 2007. 2007,

[CDL16] Colucci, L., Doshi, P., Lee, K.-L., Liang, J., Lin, Y., Vashishtha, I., Zhang, J., and Jude, A. Evaluating Item-Item Similarity Algorithms for Movies. Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM. 2016,

[CS13] Clough, P. and Sanderson, M. Evaluating the performance of information retrieval systems using test collections. Information Research 18.2 (2013).

[CYY16] Chin, W.-S., Yuan, B.-W., Yang, M.-Y., Zhuang, Y., Juan, Y.-C., and Lin, C.-J. LIBMF: a library for parallel matrix factorization in shared-memory systems. The Journal of Machine Learning Research 17.1 (2016),

[CZC15] Chang, S., Zhou, J., Chubak, P., Hu, J., and Huang, T. S. A space alignment method for cold-start TV show recommendations. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI. 2015,

[CZG16] Chen, Y., Zhao, X., Gan, J., Ren, J., and Hu, Y. Content-based top-n recommendation using heterogeneous relations. Australasian Database Conference. Springer. 2016,

[DDV14] Dooms, S., De Pessemier, T., Verslype, D., Nelis, J., De Meulenaere, J., Van den Broeck, W., Martens, L., and Develder, C. OMUS: an optimized multimedia service for the home environment. Multimedia tools and applications 72.1 (2014),

[DGL11] Dolnicar, S., Grün, B., and Leisch, F. Quick, simple and reliable: Forced binary survey questions. International Journal of Market Research 53.2 (2011),

[DM03] Delone, W. H. and McLean, E. R. The DeLone and McLean model of information systems success: a ten-year update. Journal of management information systems 19.4 (2003),

[DM92] DeLone, W. H. and McLean, E. R. Information systems success: The quest for the dependent variable. Information systems research 3.1 (1992),

[DWE05] Downie, J., West, K., Ehmann, A., and Vincent, E. The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview. 6th Int. Conf. on Music Information Retrieval (ISMIR). 2005,

[EGG12] Estes, Z., Gibbert, M., Guest, D., and Mazursky, D. A dual-process model of brand extension: taxonomic feature-based and thematic relation-based similarity independently drive brand extension evaluation. Journal of Consumer Psychology 22.1 (2012),

[ELK11] Ekstrand, M. D., Ludwig, M., Konstan, J. A., and Riedl, J. T. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. Proceedings of the Fifth ACM Conference on Recommender Systems. RecSys '11. Chicago, Illinois, USA: ACM, 2011,

[FGM15] Fisher, A. V., Godwin, K. E., Matlen, B. J., and Unger, L. Development of Category-Based Induction and Semantic Knowledge. Child development 86.1 (2015),

[GNZ07] Grassi, M., Nucera, A., Zanolin, E., Omenaas, E., Anto, J. M., and Leynaert, B. Performance Comparison of Likert and Binary Formats of SF-36 Version 1.6 Across ECRHS II Adults Populations. Value in Health 10.6 (2007),

[GRF11] Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. MyMediaLite: a free recommender system library. Proceedings of the fifth ACM conference on Recommender systems. ACM. 2011,

[HK16] Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5.4 (2016),

[JDE07] Jones, M. C., Downie, J. S., and Ehmann, A. F. Human Similarity Judgments: Implications for the Design of Formal Evaluations. ISMIR. 2007,

[JLC09] Jagadeesan, A. P., Lynn, A., Corney, J. R., Yan, X., Wenzel, J., Sherlock, A., and Regli, W. Geometric reasoning via internet crowdsourcing. 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling. ACM. 2009,

[JW02] Jeh, G. and Widom, J. SimRank: a measure of structural-context similarity. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2002,

[Kar01] Karypis, G. Evaluation of item-based top-n recommendation algorithms. Proceedings of the tenth international conference on Information and knowledge management. ACM. 2001,

 ISBN 978-80-86943-41-1

[Kel09] Kelly, D. Methods for evaluating interactive information retrieval systems with users. Foundations and Trends in Information Retrieval 3 (2009),

[KGA16] Kleiman, Y., Goldberg, G., Amsterdamer, Y., and Cohen-Or, D. Toward semantic image similarity from crowdsourced clustering. The Visual Computer 32.6-8 (2016),

[McF12] McFee, B. More like this: machine learning approaches to music similarity. PhD thesis. University of California, San Diego, 2012.

[MG12] Mirman, D. and Graziano, K. M. Individual differences in the strength of taxonomic versus thematic relations. Journal of experimental psychology: General 141.4 (2012),

[MMN02] Melville, P., Mooney, R. J., and Nagarajan, R. Content-boosted Collaborative Filtering for Improved Recommendations. Eighteenth National Conference on Artificial Intelligence. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, 2002,

[MRS08] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[Nie08] Nielsen, J. Bridging the designer-user gap (2008).

[NK11] Ning, X. and Karypis, G. Slim: Sparse linear methods for top-n recommender systems. Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE. 2011,

[OD15] Organisciak, P. and Downie, J. S. Improving Consistency of Crowdsourced Multimedia Similarity for Evaluation. Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM. 2015,

[PJH14] Pirasteh, P., Jung, J. J., and Hwang, D. Item-based collaborative filtering with attribute correlation: a case study on movie recommendation. Intelligent Information and Database Systems. Springer, 2014,

[QLX10] Qin, T., Liu, T.-Y., Xu, J., and Li, H. LETOR: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval 13.4 (2010),

[SAC07] Smucker, M. D., Allan, J., and Carterette, B. A comparison of statistical significance tests for information retrieval evaluation. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM. 2007,

[SF12] Sullivan, G. M. and Feinn, R. Using effect size-or why the P value is not enough. Journal of graduate medical education 4.3 (2012),

[Sin01] Singhal, A. Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 24.4 (2001),

[SKK01] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web. WWW '01. Hong Kong, Hong Kong: ACM, 2001,

[SPU02] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. Methods and metrics for cold-start recommendations. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM. 2002,

[Tre00] Trewin, S. Knowledge-based recommender systems. Encyclopedia of library and information science 69.Supplement 32 (2000),

[VBK14] Vargas, S., Baltrunas, L., Karatzoglou, A., and Castells, P. Coverage, Redundancy and Size-awareness in Genre Diversity for Recommender Systems. Proceedings of the 8th ACM Conference on Recommender Systems. RecSys '14. Foster City, Silicon Valley, California, USA: ACM, 2014,

[Voo01] Voorhees, E. M. The philosophy of information retrieval evaluation. Workshop of the Cross-Language Evaluation Forum for European Languages. Springer. 2001,

[VSR12] Vig, J., Sen, S., and Riedl, J. The tag genome: Encoding community knowledge to support novel interaction. ACM Transactions on Interactive Intelligent Systems (TiiS) 2.3 (2012),

[WAL17] Wang, C., Agrawal, A., Li, X., Makkad, T., Veljee, E., Mengshoel, O., and Jude, A. Content-Based Top-N Recommendations With Perceived Similarity. IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2017.

[WB99] Wisniewski, E. J. and Bassok, M. What makes a man similar to a tie? Stimulus compatibility with comparison and integration. Cognitive Psychology 39.3 (1999),

[WBM10] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD birds 200 (2010).

[WKB14] Wilber, M. J., Kwak, I. S., and Belongie, S. J. Cost-effective hits for relative similarity comparisons. Second AAAI Conference on Human Computation and Crowdsourcing. 2014.

[ZMK05] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. Improving Recommendation Lists Through Topic Diversification. Proceedings of the 14th International Conference on World Wide Web. WWW '05. Chiba, Japan: ACM, 2005,

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

Fish Motion Capture with Refraction Synthesis

Klaus Müller, Jan-Marco Hütwohl and Klaus-Dieter Kuhnert
Institute of Real-Time Learning Systems
Department of Electrical Engineering & Computer Science, University of Siegen, Germany
klaus.mueller@uni-siegen.de

Stefanie Gierszewski and Klaudia Witte
Research Group of Ecology and Behavioral Biology
Institute of Biology, University of Siegen, Germany
gierszewski@chemie-bio.uni-siegen.de

ABSTRACT

3D fish animations become more and more popular in fish behavioral research. It empowers the experimenter to design fish stimuli and their specific behavior to the experiment's needs. The fish animation can be done manually or derived from video footage. Especially automatic fish model parameter recovery for 3D animations is not well studied yet. Here we present a novel, flexible method for this purpose. It can be used to recover position, pose, bone rotation and size from single or multiple view and for single or multiple fish. Additionally we implement a novel method to compensate the fish tank's refraction effect and show that this method can decrease the error up to 80 %. We successfully applied the proposed method to two different data sets and recovered fish parameters out of single- and double-view video stream. A video attached to this paper demonstrates the results.

Keywords

motion capture, pose recovery, analysis-by-synthesis, refraction compensation, fish tracking

1 INTRODUCTION

The use of virtual 3D fish stimuli is the current trend in fish behavior research and partly replace the use of fish video or live stimulus fish partly [WGCT17]. In such kind of experiments screens with different 3D fish animations are placed next to a fish tank. Each animation shows different fish (one or several) with different appearance (e.g. skin texture or coloration), size, and morphology or behavior pattern. Inside the real fish tank are one or several test fish, which show their interest to a stimulus by physical presence in front of the corresponding screen. In order to create stimulus animations some open source software tools e.g. *FishSim Animation Toolchain*[1] or *AnyFish*[2] came into the market and help inexperienced users to create photorealistic 3D fish models and animations. The animation part of the stimulus is mostly done manually (or in case of [MSH+17] semi-automatic) since these tools do not provide methods to derive actions and behavioral patterns automatically from video footage.

In this paper we present a novel method which automatically recovers 3D fish model parameters like position, orientation and joint configuration out of single or multiple view video footage by using a model-based analysis-by-synthesis approach [Pop07]. This method was originally applied for pose recovery and tracking of humans [PMBH+10] or for human pose recovery out of a single image [KKTM15]. Especially for the task presented here this method is very promising for two main reasons: first, the time-consuming process of model-creation, which is needed for such a method, can be omitted, since the 3D fish model is already available. Second, fish have a very simple kinematic bone structure, which minimize the risk of misconvergence, what in general can happen by using this method. Here we extended this method by refraction synthesizing, which appears at the air-water border of the fish tank. Additionally we add an occlusion handling for fish. The method is based on single or multiple view silhouettes of live fish, which are approximated by view-depended artificial silhouettes, extracted out of the provided 3D fish model. For approximation we employed a least-squares method. We finally validated the presented method with video footage of single camera and dual camera, showing a single fish or a pair of fish. We annotated a video sequence of 1000 frames manually and compare this dataset to the result of the proposed algorithm (with and without refraction compensation, single and multiple view). It could be shown that the method recovered fish position and pose very precisely. Especially the refraction compensation improved the position recovery significantly. A video showing the results of the method is attached to this paper. In summary, the

[1] https://bitbucket.org/EZLS/fish_animation_toolchain for more information see [MSH+17] and [GMS+17] for its validation

[2] https://github.com/anyFish-Editor/anyFish-2.0 for more information see [VIC+13, IAW+15]

work presented in this paper solves common problems in research on fish behavior and contributes the following features:

- precise transfer of fish movement patterns from video footage to a photo-realistic 3D fish model (as used in skeletal animations) precisely
- high position precision based on refraction compensation by synthesis during optimization
- very flexible: single or multiple view camera setup, single or multiple fish, fast (without refraction compensation) or precise

The paper is divided into six chapters. In chapter 2 we present related work. This is followed by chapter 3 giving information regarding preliminaries. In Chapter 4 the method is described in detail. We present in chapter 5 the results of the method which are finally discussed in chapter 6.

2 RELATED WORK

Most motion capture research has been done and is still going on in the field of human motion capture. There are several different methods available: on the one hand wearable motion trackers are used, which track the position and rotation of single joints (head, arms, legs etc.) (see e.g. [RLS09]). On the other hand there are optical motion capture methods. These are divided in system which use markers mounted to the human body and markerless systems, which use single or multiple RGB-cameras (e.g. [ST02, PMBH+10]) or RGB-depth cameras (see e.g. [SSK+13]). Nowadays there are also deep learning methods used for pose recovery like presented in [CSWS17] or in [WRKS16]. In contrast to human motion capture, fish pose recovery leads to special challenges: firstly, the use of motion trackers or markers for visual pose recovery is very difficult, since wearable motion trackers are not available for small fish or markers can only be fixed under great difficulties to fish. Secondly, RGB-D cameras (e.g. Microsoft Kinect), which pushed the human pose recovery research forward massively, can only be used very limited: such cameras mostly use active light, which brings difficulties regarding reflection and refraction while light travels through different media (e.g. water and air). Due to these facts multiple view camera setups are the most used configuration for 3D tracking and pose recovering of fish. Besides some research in field of fish position tracking in 2D and 3D (for a review see [DDYP13]), there is little research in fish pose-recovery. Takahashi et al. introduce a method to extract fish position and posture from orthogonal video footage. They used a simple 3D fish model, which was projected to the real images. With the help of a brute force, box constrained search algorithm they estimated the model-parameters in a way, that the projection fits best to the recorded fish images. They finally used the gathered motion data to estimate a locomotion model of the fish [THHN00]. Butail and Paley estimated 3D position and shape to analyse fish schooling kinematics [BP10]. They modelled the fish shape as bendable ellipsoid. Based on this model fish pose, position and bending is estimated out of 2D silhouettes with the help of a particle filter. Later on they improved this method and extended the used 3D-model [BP12]. In contrast to the former model, the newer model consisted of estimated cross-sectional ellipses, which were ordered along a three-dimensional midline, describing the bending of the fish body more precisely. They used simulated annealing to match 2D silhouettes to the model and to find the best model parameter set. The cost function is based on the sum of distances between occluding contour points and the model surface. Voesenek et al. used a similar but more precise model with more degrees of freedom regarding fish bending and rotation [VPvL16]. They also approximated 2D silhouettes to a 3D-model, which consists of merged ellipsoids along the longitudinal axis. For finding the optimal model-parameters they re-projected the model to the virtual cameras and calculated a scalar value describing the overlap and used a downhill simplex algorithm for optimization. Besides extraction of fish motion they used the system to derive resultant forces and torques of fish during swimming.

In contrast to the former work, the proposed method differs in the following:

- motion capture for 3D fish animation: this method uses a 3D fish animation model with bones to recover position, pose and bending. The resulting parameter set can directly be used to animate 3D-models
- the proposed method synthesizes the refraction caused by the air-water border
- we use a non-linear least-squares method to approximate the fish position, pose and bending, which uses all silhouette pixels separately for optimization
- the method is very flexible and can be used for single or multiple fish, for single- or multiple camera setups, precise (with refraction compensation) or fast (without refraction compensation)

3 PRELIMINARIES

3.1 Calibration

Since our method is specialized for fish in aquaria, we use an easy and precise calibration method, which was especially developed for this purpose (see [MSKK14]). The method assumes, that camera position and alignment are static in relation to the aquarium. Based on

 ISBN 978-80-86943-41-1

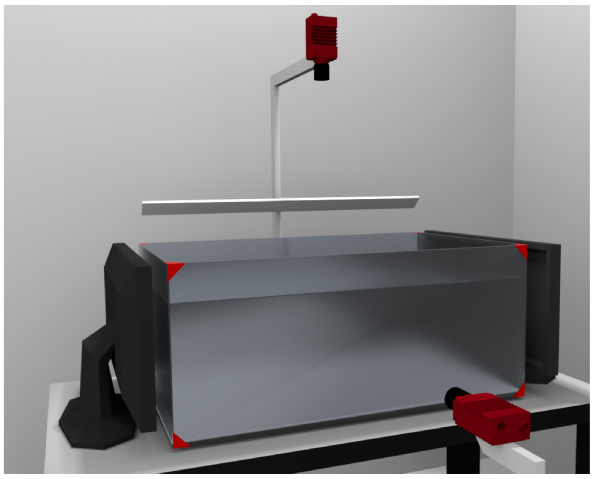


Figure 1: Setup with two cameras. The red corners at the tank are used for automatic-calibration of extrinsic camera-parameters and position and normal calculation of the fish-tank sides.

markers mounted to the corners of the fish-tank, an optimization algorithm estimates the camera parameters with respect to the aquarium. Besides the extrinsic camera-parameter estimation the method also estimates the normals and positions of the tank windows automatically and the normal and the position of the water surface semi-automatically. This information is used for the refraction calculations afterwards.

3.2 Contour retrieving from video

Since the proposed method is specialized for aquaria, we consider scene and cameras as static. This provides the opportunity to use classical background subtraction methods to divide the camera image in background (tank) and foreground (fish). Based on this binary image we extract the silhouette S of all foreground objects. In case of using multiple cameras fish silhouettes originated by mirroring in the tank's glass walls can be detected and eliminated by using epipolar geometry constraint. We also use this constraint to assign silhouettes of fish in multiple views (see also [MSKK14]).

3.3 3D-model

The pose recovery result strongly depends on the 3D fish-model quality: the higher the shape similarity between the 3D-model and the live fish is, the better the model can be approximated. Depending on the fish species, it can be enough to use two different models for male and female, like done in the validation presented here with sailfin mollies (*Poecilia latipinna*). In case of fish species varying strongly within sex it could be necessary to use several different models for female and male, mapping its variation. In general, the fish model has to be implemented a as 3D-mesh. Even thin parts of the fish like fins have to be designed as 3D object in order to retrieve the contour by the proposed algorithm presented in section 4.1. Since the proposed method is based on fish silhouettes, texture and reflection properties of the model can be ignored. In order to deform the fish model, all parts of the mesh whose position and rotation should be recovered by the algorithm have to be connected to the skeleton bones. In the field of computer animation, this type of model presentation and animation is called *skeletal animation*. For the validation presented in this paper we developed fish models with the help of the free available fish designer included in *FishSim Animation Toolchain* [3]. At the moment the toolchain includes five different fish species (the sailfin molly *Poecilia latipinna*, the Atlantic molly *Poecilia mexicana*, the guppy *Poecilia reticulata*, the three-spined stickleback *Gasterosteus aculeatus*, and a chichlid *Haplochromis* spp.). In general it is possible to add new fish species to the toolchain. More information on the toolchain can be found in [MSH+17] and see also [GMS+17] for a validation of the generated fish stimuli in research.

4 POSITION, POSE AND SKELETON RECOVERY

The proposed method aims to approximate the former described 3D-model to the live fish in tank. The approximation is based on silhouettes of live fish, captured from single or multiple cameras. The silhouette of live fish is extracted as described in section 3.2 and is compared to the artificial contour of the 3D-model. In order to create as similar artificial silhouettes as possible it is necessary to adjust and position the virtual cameras exactly like the live ones. In order to do so, we use the estimated calibration information as described in section 3.1. Besides the camera parameters the dimensions of the fish are necessary to approximate the model parameters. Therefore we also implement a method which estimates fish size in a preprocessing step from single- or multiple view video sequences (see section 4.6). With the help of the in 4.1 described method the silhouettes are extracted from the 3D-model and are compared to real silhouettes by an error function. An optimization algorithm, described in section 4.4, optimizes the model parameters in order to minimize the error function (see section 4.3). To lower the risk of converging to a local minimum of the error function, we apply an initialization method described in section 4.5 at the beginning of a video sequence. Since the air-water boarder of the fish tank causes refraction effects, we implement a compensation method in order to increase the accuracy (see section 4.2). The whole workflow is shown figure 2.

[3] `https://bitbucket.org/EZLS/fish_animation_toolchain`

ISBN 978-80-86943-41-1

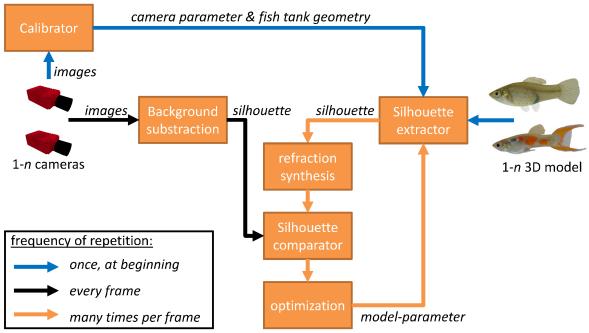


Figure 2: Schematic of the fish recovering system.

4.1 2D silhouette extraction from 3D-model

There are several different ways to extract 2D silhouettes from a 3D-model: one obvious possibility is the use of a render system as included in *FishSim Animation Toolchain*. The silhouette is rendered to a homogeneous surface and can easily be extracted by thresholding. In practice, it turned out that the rasterization which is used by the render system to draw primitives to a pixel-based device or image, is too imprecise since its resolution is limited to the pixel resolution of the underlying device. Especially during the optimization process this fact causes that the algorithm does not converge correctly. For that reason we apply the classical method leaned on [BS00] to retrieve silhouettes from polygonal mesh structures. This method is based on the principle that a silhouette edge will appear if one front-face (front side of mesh-triangle) of two neighboring mesh triangles is visible and the other one is invisible for the camera. In order to find silhouette edges we traverse all k polygon mesh triangles and check if the normal $\vec{n}_k$ of triangle $\triangle ABC_k$ is pointing in the same direction as the viewing vector $\vec{v}$ of the camera and store the result in a new vector $\vec{f}$:

$$f_i = \begin{cases} 1 & \text{if } \vec{v} \circ \vec{n}_i \geq 0 \quad i \in k \\ 0 & \text{if } \vec{v} \circ \vec{n}_i < 0 \quad i \in k \end{cases} \tag{1}$$

In case that f_i and f_j of neighboring triangles $\triangle ABC_i$ and $\triangle ABC_j$ are different the shared edge of both triangles is part of the silhouette. The gathered 3D silhouette points are converted with the help of the camera projection matrix to a 2D pixel coordinates $\tilde{S}$. Depending on the 3D-mesh resolution and the camera distance to the object, it can happen that the distance between two neighboring silhouette pixels is several pixel units. Especially for the comparator algorithm (see 4.3) that searches nearest neighbor pixels between real and virtual silhouette this could cause bigger errors. For that reason, we interpolate silhouette pixels in case that the distance is bigger than one pixel unit between neighboring silhouette pixels. Finally, this method extracts a vector of silhouette pixels $\tilde{S}_X$ out of the 3D-mesh model $M(X)$ which is based on the model parameters X.

4.2 Refraction compensation

Since the proposed method is specialized for pose recovery of fish in aquaria we also address the problem of refraction. Especially for stereo or multiple view camera setups refraction can cause errors of several centimeters [MSKK14] and hinder the mapping of silhouettes from different views. In general, we calculate the pixel ray refraction with the help of Snell's law, which calculates the refraction angle of the ray with the help of media's refractive indexes (n_1, n_2) and the incident angle (angle between surface and ray).

$$\frac{n_1}{n_2} = \frac{sin(\beta)}{sin(\alpha)} \tag{2}$$

In contrast to a ray-tracing approach, in which the optical path of each pixel ray can be calculated step-by-step, we have to go the way around: we start at the 3D coordinate of the silhouette edge and have to find the intersection point with the refraction plane (fish tank plane) in order to calculate the refraction angle and finally the 2D pixel coordinate. At first an additional plane P is calculated, which is defined by the normal vector $\vec{n}$ of the intersection plane I (aquarium window or water surface defined during calibration), the camera position vector $\vec{c}$ and the position vector of a silhouette point $\vec{s}_i$ ($i \in$ all 3D silhouette points). Next, the line of intersection $g(x) : \vec{x} = \vec{o} + x\vec{r}$ between intersection plane I and plane P is calculated. There are several algorithms available to calculate the line of intersection between two planes and therefore it is here not discussed further. The intersection point of the pixel ray lays on this line. With the help of the law of sines we can calculate α and β (see figure 3):

$$sin(\alpha) = \frac{\|(\vec{x} - \vec{c}) \times \vec{n}\|}{\|(\vec{x} - \vec{c})\| \cdot \|\vec{n}\|} \tag{3}$$

$$sin(\beta) = \frac{\|(\vec{x} - \vec{s}_i) \times \vec{n}\|}{\|(\vec{x} - \vec{s}_i)\| \cdot \|\vec{n}\|} \tag{4}$$

In order to find the final intersection point on the line g, we combine the equations (2), (3) and (4) and minimize it:

$$\min_{x \in \mathbb{R}} \frac{\|(g(x) - \vec{c}) \times \vec{n}\|}{\|(g(x) - \vec{c})\| \cdot \|\vec{n}\|} \frac{\|(g(x) - \vec{s}_i)\| \cdot \|\vec{n}\|}{\|(g(x) - \vec{s}_i) \times \vec{n}\|} - \frac{n_1}{n_2} \tag{5}$$

In order to initialize the minimization well, the initial value x_i is defined by calculating the intersection point D (position vector $\vec{d}$) between $\overline{S_iC}$ and intersection plane I. D also lays on g and the factor x_i is calculated as follows:

$$x_i = \begin{cases} \frac{o_x - d_x}{r_x} & \text{if } r_x \neq 0 \text{ else} \\ \frac{o_y - d_y}{r_y} & \text{if } r_y \neq 0 \text{ else} \\ \frac{o_z - d_z}{r_z} \end{cases} \qquad \text{with } [\vec{d}, \vec{o}, \vec{r}] = \begin{pmatrix} [d,o,r]_x \\ [d,o,r]_y \\ [d,o,r]_z \end{pmatrix} \tag{6}$$

ISBN 978-80-86943-41-1

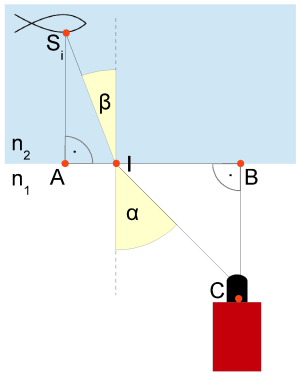


Figure 3: Refraction. A, B and I lie on the intersection plane, which is also the separation plane between air and water. A ray starts at the fish's surface S, intersects in I and gets refracted. Finally it hits the camera center C.

4.3 Contour comparator

To approximate pose and position of fish, we compare the view depending silhouettes of live and virtual fish. This is done by searching the nearest virtual silhouette pixel $v_i \in S_v$ for each real silhouette pixel $r_i \in S_r$ in 2D pixel space. We employ a brute force matching algorithm which searches the nearest neighbor on base of the Euclidean distance. We finally store all distances in order of real pixel silhouettes. In case of a single camera setup the error vector has always the same size as the real silhouette and depends on the model parameters X.

$$e(X) = \begin{pmatrix} e_0(X) \\ e_1(X) \\ \vdots \\ e_n(X) \end{pmatrix} = \begin{pmatrix} \|r_0 - v_i\| \\ \|r_1 - v_j\| \\ \vdots \\ \|r_n - v_k\| \end{pmatrix} \quad v_n \in \tilde{S}_X \quad (7)$$

In case of a multiple camera setup with more than one silhouettes of a single live fish (one silhouette for each camera view), the error vectors for each silhouette are stacked in a new error vector. $e^k(X)$ describes the error vector $e(X)$ of camera k. The final error vector of a multiple view setup has the same size as the sum of all real silhouettes pixels.

$$e(X) = \begin{pmatrix} e^1(X) \\ e^2(X) \\ \vdots \\ e^k(X) \end{pmatrix} \quad (8)$$

4.4 Optimization

In order to approximate the pose and position parameters of the virtual fish we employ a least-squares optimization method (combination of Levenberg-Marquardt method and quasi-newton method [DJGW81]) which optimizes the model parameters by reducing the error-vector described in equation 7. The optimization method uses the derivative of the error function with respect to the parameter vector X. We numerically approximate a derivative vector for each real silhouette pixel r_i as follows:

$$D_i(X) = \begin{pmatrix} \frac{e_i(X_{0+\varepsilon}) - e_i(X_{0-\varepsilon})}{2\varepsilon} \\ \frac{e_i(X_{1+\varepsilon}) - e_i(X_{1-\varepsilon})}{2\varepsilon} \\ \vdots \\ \frac{e_i(X_{j+\varepsilon}) - e_i(X_{j-\varepsilon})}{2\varepsilon} \end{pmatrix} \quad (9)$$

$$with\ X_{0+\varepsilon} = \begin{pmatrix} x_0 + \varepsilon \\ x_1 \\ \vdots \\ x_j \end{pmatrix}$$

In case of a multiple camera setup, the derivative has to be calculated for all silhouette pixels of all views. In case of using a single camera setup it is recommended to use a Kalman filter to stabilize the fish position, pose and bending.

4.5 Initialization

The better the optimization initialization the higher the probability of convergence and the faster the optimization can be finished. Especially for real-time applications fast initialization is very important. For this task, we employ a method which searches an initial model parameter set out of a database based on simple silhouette features like, position of snout, centroids of silhouette quarters or angle of major segment axis. In order to find the fitting model parameter set for the incoming silhouettes, the features are extracted and a brute force matcher searches the nearest neighbor. To speed up the process of pose initialization we use the method described in [MSK16] which defines a feature subset regarding the pose-space location. In order to initialize the position of fish, we calculate the centroid of the live fish silhouette and define the pixel-ray of this centroid pixel with the help of the camera projection matrix. In case of an multiple camera setup, we calculate the rough intersection point of these pixel-rays and use it as initial position. If a single view setup is used, the approximated position is calculated by shifting the model along the centroid pixel ray to the middle of the fish tank. In case of a post processing application, it is also possible to find the initial pose and position manually.

4.6 Estimation of fish-size from single and multiple view imagery

Besides a fish's shape, its correct size is very important for a precise recovery of position and pose of the live

ISBN 978-80-86943-41-1

fish. One option is to measure the size of the fish manually. This can be quite difficult since the fish has to be caught and its body has to be aligned along the measurement tool. An easier option is to use the computer vision system to measure the size of the fish. In the proposed method we also apply the in subsection 4.4 described optimization method in a preprocessing step. To do so we record a short video sequence of the swimming fish and besides the pose and position parameters we also optimize the size in x-, y- and z-direction. Depending on the used model, it is also possible to optimize the scale of the bones in order to adjust the shape of the fish automatically. We average the size parameters over the whole test sequence and use these parameters for the actual recovery process. For a multiple view setup, the size can be approximated fast and precisely. In contrast, in a single-view setup the model-size can not be recovered exactly: the projected size of the silhouette depends on the size of the model as well as on the distance between camera and object. For that reason we use a constrained size optimization, in which the fish position is bounded to the size of the fish tank. In order to get a good result, the recorded fish movement should cover the area in front of the tank's front and back wall.

4.7 Multiple fish and occlusion handling

The method presented in this paper is capable of multiple fish tracking. It is recommended to use a multiple camera setup in order to increase the stability of the system in case of occlusion. As long as no occlusion occurs, every fish can be handled separately according the previous described method. In case of occlusion, we modify the method as follows:

Silhouette mapping

Since the mapping of silhouette and fish is straightforward in case of a single fish (single silhouette to single fish), the problem of silhouette mapping occurs if several fish have to be tracked. In order to find the right silhouette for each fish, we compare the extracted contours of the current image regarding equation 7 with the silhouette of each fish model of the last frame. We assign the extracted contour to the model with the smallest error. This is done for each frame and for each camera view.

2D silhouette retrieving for occluding fish

If two or more fish cover each other in a camera view, the background subtraction method will just provide a single silhouette for these fish. For approximation of the virtual contour to the real silhouette it is necessary to reconstruct the silhouette as good as possible. We do so by creating the silhouette of each involved fish separately and merge these silhouettes together. This results in a single silhouette which consists of all outer silhouette edges.

Optimization in case of occlusion

Due to the fact that more fish are involved in the optimization process we combine all parameter vectors X of the involved fish to a new parameter vector. The contour comparator works the same way as described in section 4.3 except that the silhouette pixels of the combined silhouette are used for the camera view where the occlusion takes place. For the optimization all silhouette pixels of all fish in all camera views were used to approximate the virtual models to the live ones. Tests showed that the combined silhouette of multiple fish brings a higher risk of wrong convergence. For that reason we will check if the involved fish has a separate silhouette in another camera view and push this silhouette twice to the optimization process. By doing so this fish silhouette has a higher impact to the optimization process and the risk of wrong convergence decreases.

4.8 Handling of transparent fish parts

Another difficulty of fish pose recovery is the handling of transparent parts like fins. In our experiments we figured out that especially semi-transparent fins can cause trouble: depending on the fish position and alignment, it could happen that, for the background subtraction system, a fin is visible in some regions of the fish tank and invisible in other regions. This can cause problems for the method presented here since we extract (see chapter 4.1) the outer silhouette of the fish. If for example the caudal fin is not always visible, it will influence the optimization algorithm negatively. In order to handle this problem, we recommend to organize fish parts (e.g. fins) in mesh-groups. If a part is not detected by the background subtraction, it can be easily removed from the model. In case a fin is detected from time to time we extract the silhouette of this fin separately and add it to the total contour. By doing so both contours (with and without fin) are available and the contour comparator searches for the best matching one. This is also shown in figure 4.

5 RESULTS

We compared the results of the here introduced method regarding runtime and precision with a manually annotated dataset. This included the results of single-camera setup, multiple camera setup, with and without refraction compensation. Additionally, we also applied the method to a dataset of two fish including occlusion in one and both camera views.

5.1 Dataset

The dataset consisted of 1000 manually annotated frames which show a single female sailfin molly swimming in a fish tank (26 cm x 18 cm x 17 cm). The fish had a length of approximately 5 cm which corresponds to about 180 to 200 pixels. For annotation we manually

 ISBN 978-80-86943-41-1

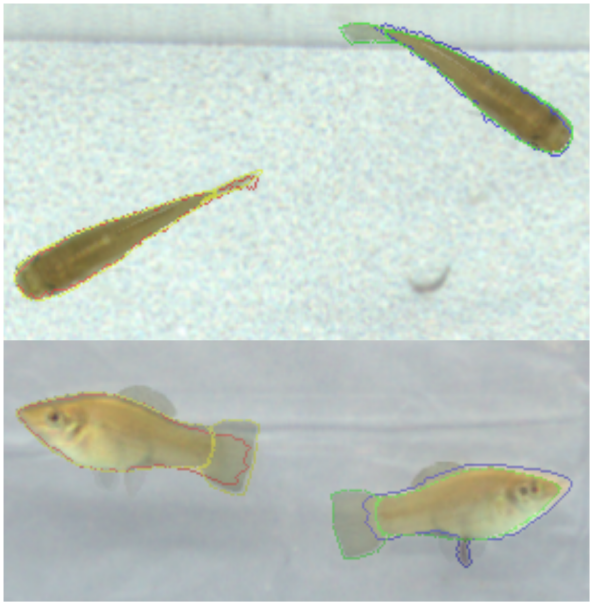


Figure 4: Silhouettes of two fish. Silhouettes of background subtraction is marked blue and red, the virtual silhouettes yellow and green. The quality of the real contour was not stable at all. Especially the transparent fins were sometimes not detected by the background subtraction like in case of the right fish.

adjusted the model parameters of the according 3D fish-model and rendered (raytracing with refraction based on the method validated in [MSKK14]) it frame-by-frame over the video footage of two cameras, observing the fish. The second dataset showed two female sailfin mollies (about 5 cm, approx. 120 to 140 pixels in length), swimming close to each other through a bigger fish tank (60 cm x 30 cm x 30 cm). Since the fins of the used fish are nearly transparent, the background segmentation method sometimes did not detect the whole fin. This caused an imprecise silhouette (see figure 4) and special demands on the proposed method regarding fin pose recovery. Both datasets were recorded by two cameras (Allied Vision Technologies, Prosilia GT1910c) with a resolution of 1920 x 1080 pixels and with a frame rate of 57 frames per second mounted above and in front of the tank. The cameras were synchronized by hardware trigger.

5.2 Model

The used 3D-model was created with the *FishSim Animation Toolchain* and consisted of 946 vertices and 36 bones. For the optimization we just used 17 bones: head, four backbones and twelve bones of the tail (caudal) fin. In order to decrease the model parameters we used a function that approximates the fish bending to a single bending value (see [SMK15]). In total the optimization process comprised six parameters per fish: position (x,y,z), rotation (pitch, yaw) and bending. During size estimation the parameter set was extended by three size parameters. We also tested the method with three and eight different bending parameters.

5.3 Initialization

Before the optimization process started a rough initial position and rotation of fish were found as described in section 4.5. The used database consisted of 32400 parameter sets. The features were extracted in a preprocessing step out of 32400 artificial fish images, showing a single fish (rendered from the used 3D-model) rotated in two degree steps around the main axes. Fish size was estimated out of 100 images from two perspectives (top and front) using the method described in section 4.6. We estimated the size along x-, y- and z- axis of fish separately. We also estimated the size manually. The results of manual and automatic size estimation differed very slightly.

5.4 Refraction compensation

We applied the proposed method to the first dataset, showing a single fish from two perspectives (front and top view), and recovered fish pose, position and bending with and without refraction compensation. The recovery results especially differed in terms of position error. The method using refraction compensation reached a mean error of 1.13 mm (standard deviation 1.00 mm, biggest error 3.9 mm) and the version without compensation reached a mean error of 8.24 mm (std. dev. 4.91 mm, biggest error 19.5 mm). Regarding rotation and bending error the methods differed less, but the version with refraction compensation was always better (see figures 5, 6, 7). This is due to the fact, that the silhouettes of the not refracted version do not fit precisely to each other, especially at the outer borders of the fish tank where the refraction is particularly high.

5.5 Single view vs. dual view

Besides the dual camera setup, we also applied the method with refraction compensation to single camera setup. We used the first dataset and applied the method separately to the top and front camera. As expected, the errors of position, rotation and bending increased. Especially the error for the dimension along the direction of camera view increased strongly (see figures 5, 6, 7). In general, for the here used fish species (sailfin molly) the top view camera delivered better results. This was mainly due to the fact, that the bending can better be observed from top (see figure 4). For experiments with a narrow third dimension (little water or thin tank) this method can be a cost-effective and less computation-intensive alternative to the multiple camera setup.

5.6 Occlusion of multiple fish

We recovered pose and position of two fish out of the second dataset (dual camera setup). The recorded video

ISBN 978-80-86943-41-1

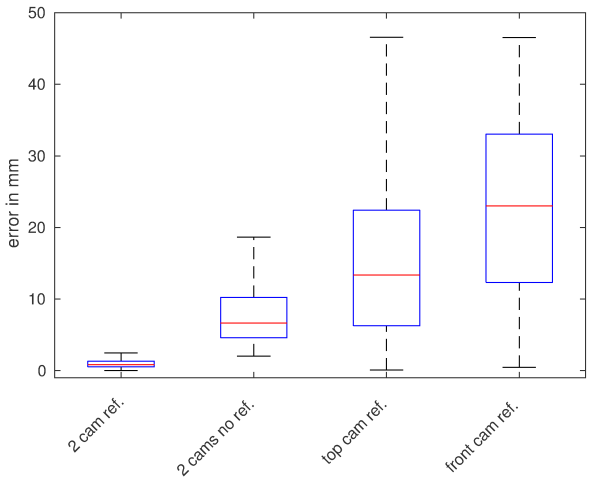


Figure 5: Fish position error split by method of application. The diagram shows box-plots, which indicates the median (red line) the 25th and 75th percentiles(blue box) and the biggest error values (whiskers) not considered outliers. From left to right it shows the position error (in relation to ground truth in mm) 1. using two cameras and refraction compensation 2. two cameras without refraction compensation 3. single top view camera with refraction compensation and 4. single front view camera with refraction compensation.

included several sequences with fish occlusion in one or both views. The algorithm recovered fish position and pose of both reliably. If an occlusion occurred in all views, it can happen that the algorithm will converge to the wrong fish model. This problem can be minimized by applying a kalman-filter to each fish. In figure 8 an occlusion case and the recovered fish positions and poses are shown. Additionally, a video sequence showing occluding situations can be found inter alia in

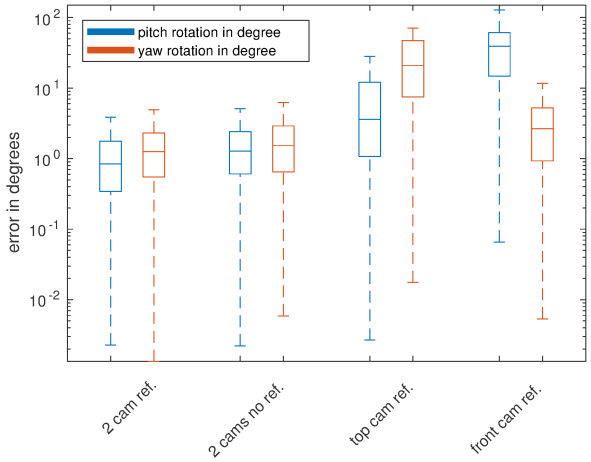


Figure 6: Fish pitch- and yaw-rotation error split by method of application. The diagram shows box-plots of pitch-rotation (blue) and yaw-rotation (red) in degrees. Since the used test fish nearly never rotate around roll-axis, this rotation was neglected.

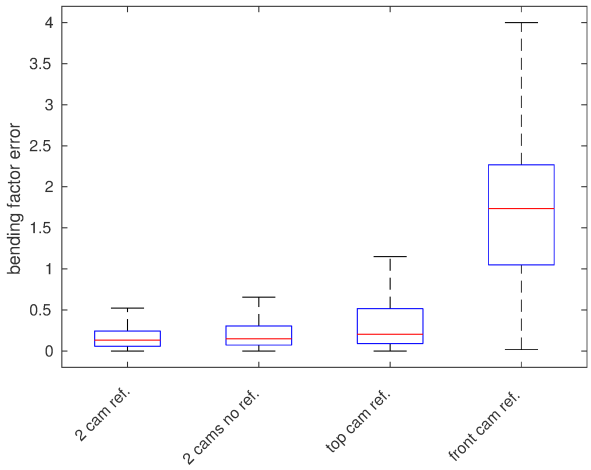


Figure 7: Fish position error split by method of application. The diagram shows box-plots of bending-factor errors. The factor maps fish bending to model parameters.

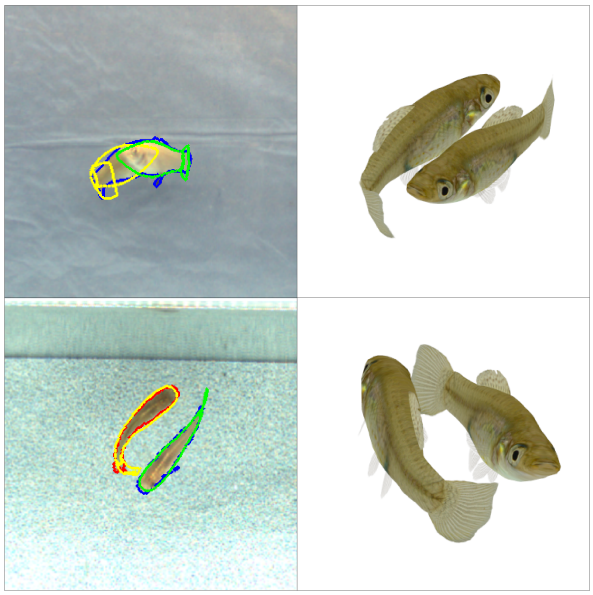


Figure 8: Reconstruction of two fish with occlusion. On the left the capture images with the silhouette overlay (blue and red contour - captured silhouettes; yellow and green contour - silhouettes of virtual fish). On the right, rendered images of the reconstructed fish models captured from two different perspectives.

the supplementary file. In general, in case of occlusion, the recovery method slightly loses accuracy.

5.7 Runtime

The software was tested on a system with Intel I7-3770 (4 x 3.4 GHz) CPU, 16 GB memory and Ubuntu 14.04 operation system. The algorithm was implemented with the help of the following open source libraries: *OpenCV* (version 2.4.12, `https://opencv.org/`) for computer vision tasks, *Dlib* (version 19.2, `http://dlib.net/`) for optimization and the game en-

ISBN 978-80-86943-41-1

Table 1: Algorithm's runtime under different configurations

configuration	runtime per frame / 2 frames in sec. (fps)
single fish, single camera, no refraction compensation	0.07 (14.2)
single fish, single camera, with refraction compensation	0.25 (4)
single fish, two cameras, no refraction compensation	0.14 (7.1)
single fish, two cameras, with refraction compensation	0.5 (2.0)
two fish, two cameras, with refraction compensation	0.7 (1.4)
single fish, two cameras, with refraction compensation and size estimation (three additional parameters)	0.67 (1.49)
single fish, two cameras, with refraction compensation and eight bending parameters	0.92 (1.08)
single fish, two cameras, no refraction compensation and eight bending parameters	0.25 (4)

gine *irrlicht* (version 1.8.1, `http://dlib.net/`) for rendering and silhouette extraction. We measured the mean time which was needed to process one frame (or two frames in case of two cameras). Table 1 gives a rough impression of the computational-intensity of different configurations. The refraction compensation was relative computational-intensive since every silhouette pixel was optimized separately. For runtime improvement it could be interesting to find an analytic solution of equation 5 which substitutes the optimization. In general it can be noted, that increasing the number of cameras and of fish the runtime increases approximately linear. Additionally, with up-to-date hardware, the method can be real-time capable.

6 CONCLUSION

In this work we introduce a new method to approximate 3D fish skeletal model parameters out of single- or multiple view video stream. We propose a new method to synthesize the refraction effect during optimization. We successfully applied the method to two different datasets with different configurations: we extracted model parameters for a one and two fish with and without refraction compensation. We showed that refraction compensation increases the recover accuracy: for position recovery the mean error was reduced by $\sim$85 % for rotation by $\sim$20 % and for bending by $\sim$11 %. We demonstrated that it is possible to recover the 3D-model parameters out of a single view video stream and reduce the runtime at the same time. By doing so it is possible to use the method in real-time application.

ACKNOWLEDGEMENTS

The presented work was developed within the scope of the interdisciplinary, DFG-funded project "virtual fish" (KU 689/11-1 and Wi 1531/12-1) of the Institute of Real-Time Learning Systems (EZLS) and the Research group of Ecology and Behavioral Biology at the University of Siegen.

7 REFERENCES

[BP10] Sachit Butail and Derek A Paley. 3d reconstruction of fish schooling kinematics from underwater video. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2438–2443. IEEE, 2010.

[BP12] Sachit Butail and Derek A Paley. Three-dimensional reconstruction of the fast-start swimming kinematics of densely schooling fish. *Journal of the Royal Society Interface*, 9(66):77–88, 2012.

[BS00] John W Buchanan and Mario C Sousa. The edge buffer: A data structure for easy silhouette rendering. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 39–42. ACM, 2000.

[CSWS17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, volume 1, page 7, 2017.

[DDYP13] Johann Delcourt, Mathieu Denoël, Marc Ylieff, and Pascal Poncin. Video multitracking of fish behaviour: a synthesis and future perspectives. *Fish and Fisheries*, 14(2):186–204, 2013.

[DJGW81] John E Dennis Jr, David M Gay, and Roy E Walsh. An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 7(3):348–368, 1981.

[GMS+17] Stefanie Gierszewski, Klaus Müller, Ievgen Smielik, Jan-Marco Hütwohl, Klaus-Dieter Kuhnert, and Klaudia Witte. The virtual lover: variable and easily guided 3d fish animations as an innovative tool in mate-choice experiments with sailfin mollies-ii. validation. *Current Zoology*, 63(1):65–74, 2017.

 ISBN 978-80-86943-41-1

[IAW+15] Spencer J Ingley, Mohammad Rahmani Asl, Chengde Wu, Rongfeng Cui, Mahmoud Gadelhak, Wen Li, Ji Zhang, Jon Simpson, Chelsea Hash, Trisha Butkowski, et al. anyfish 2.0: an open-source software platform to generate and share animated fish models to study behavior. *SoftwareX*, 3:13–21, 2015.

[KKTM15] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4390–4399, 2015.

[MSH+17] Klaus Müller, Ievgen Smielik, Jan-Marco Hütwohl, Stefanie Gierszewski, Klaudia Witte, and Klaus-Dieter Kuhnert. The virtual lover: variable and easily guided 3d fish animations as an innovative tool in mate-choice experiments with sailfin mollies-i. design and implementation. *Current Zoology*, 63(1):55–64, 2017.

[MSK16] Klaus Müller, Ievgen Smielik, and Klaus-Dieter Kuhnert. Optimal feature-set selection controlled by pose-space location. In *VISIGRAPP (4: VISAPP)*, pages 200–207, 2016.

[MSKK14] Klaus Müller, Jens Schlemper, Lars Kuhnert, and Klaus-Dieter Kuhnert. Calibration and 3d ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, pages 626–634. IEEE, 2014.

[PMBH+10] Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 663–670. IEEE, 2010.

[Pop07] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer vision and image understanding*, 108(1-2):4–18, 2007.

[RLS09] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep*, 2009.

[SMK15] Ievgen Smielik, Klaus Müller, and Klaus-Dieter Kuhnert. Fish motion simulation. In *ESM-European Simulation and Modelling Conference*, pages 392–396, 2015.

[SSK+13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[ST02] Cristian Sminchisescu and Alexandru Telea. Human pose estimation from silhouettes. a consistent approach using distance level sets. In *10th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG'02)*, volume 10, 2002.

[THHN00] Hiroki Takahashi, Junji Hatoya, Naoki Hashimoto, and Masayuki Nakajima. Animation synthesis for virtual fish from video. In *Proceedings of the 10th ICAT (International Conference on Artificial reality and Telexistence)*, pages 90–97, 2000.

[VIC+13] Thor Veen, Spencer J Ingley, Rongfeng Cui, Jon Simpson, Mohammad Rahmani Asl, Ji Zhang, Trisha Butkowski, Wen Li, Chelsea Hash, Jerald B Johnson, et al. anyfish: an open-source software to generate animated fish models for behavioural studies. *Evolutionary Ecology Research*, 15(3):361–375, 2013.

[VPvL16] Cees J Voesenek, Remco PM Pieters, and Johan L van Leeuwen. Automated reconstruction of three-dimensional fish motion, forces, and torques. *PloS one*, 11(1):e0146682, 2016.

[WGCT17] Klaudia Witte, Stefanie Gierszewski, and Laura Chouinard-Thuly. Virtual is the new reality. *Current Zoology*, 63(1):1–4, 2017.

[WRKS16] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

Perception of basic emotion blends from facial expressions of virtual characters: pure, mixed, or complex?

Meeri Mäkäräinen
Aalto University
Finland
meeri.makarainen@aalto.fi

Jari Kätsyri
Maastricht University
The Netherlands
jari.katsyri@maastrichtuniversity.nl

Tapio Takala
Aalto University
Finland
tapio.takala@aalto.fi

ABSTRACT

As animated virtual characters in games, movies and other applications become more humanlike, it becomes more and more important to be able to imitate the complicated facial behaviour of a real human. So far, facial expression animation and research have been dominated by the basic emotions view, limited to the six universal expressions: anger, disgust, fear, joy, sadness and surprise. More complex facial expressions can be created by blending these basic emotions, but it is not clear how these blends are perceived. Are they still perceived as basic emotions or combinations of basic emotions, or are they perceived as expressions of more complex emotions? We used a series of online questionnaires to study the perception of all pairwise blends of basic emotions. The blends were produced as a sum of facial muscle activations in the two basic emotions, using a physically-based, animated face model. Our main finding is that several basic emotion blends with an opposite valence are perceived as complex emotions that are neither pure emotions nor their blends. Blends of basic emotions with a similar valence are typically perceived as pure basic emotions (e.g., a blend of anger and disgust is perceived as pure anger). Only one of the blends (joy+surprise) was perceived as a blend of two different basic emotions.

Keywords

virtual agent, basic emotions, facial expressions, mixed emotions, affective computing, perception.

1 INTRODUCTION

Animated characters are used widely in games, movies and virtual applications, and the recent advances in rendering and modeling techniques have enabled them to be highly human-like. This creates pressure to develop understanding of more fine-detailed facial expressions of animated characters, to enable the development of their facial behaviour in order to keep up with the development of appearance. Facial expressions of emotion are often conceptualized as discrete expressions of *basic emotions*. Anger, disgust, fear, joy, sadness, and surprise are considered as six basic emotions with universally recognizable characteristic facial expressions [6]. Although the basic emotion view remains debated, it still remains a useful basis for facial expression research. Dimensional emotion models, such as the pleasure-arousal-dominance model have been useful in the research on emotional states and reactions, but research on the perception of facial expressions has been largely based on the basic emotion approach. In particular, the facial expressions of most virtual agents have been based on the basic emotion view [11, 9].

Although the basic emotion approach may be a good basis for modeling facial behaviour in virtual agents, a set of six separate facial expressions is very limited for producing natural facial behaviour. For a wider variety of different facial expressions, a common solution is to create composite facial expressions that combine two basic emotions into one expression. Several techniques have been introduced, including interpolation in a two-dimensional emotion space [3], displaying different emotions in the upper and lower parts [14] or left and right halves of the face [1], and additive methods [16]. In some cases, a dimensional emotion model is used where each separate emotion is considered a point in an emotional space, and each emotion is combined only with its nearest neighbours [2].

Blending two basic emotions in one facial expression can be justified because the basic emotions are not mutually exclusive. For example, happiness and sadness [12], or amusement and disgust [10], can be experienced together. *Mixed emotions*, in which two or more basic emotions are present, can produce a powerful experience, such as the pleasure of listening to sad music. It is also possible for real humans to make facial expressions that combine two basic emotions. Sometimes the facial displays of old people are interpreted as mixed emotions, because their wrinkles suggest another emo-

tion than their actual expression. Thus, mixed signals on the face are associated with mixed emotions. It is important for virtual agents to be able to express mixed emotions, as this increases the perceived empathy of the agent [15]. However, it is unclear whether the perception of blended expressions of virtual characters can be better understood as a perception of two simultaneous basic emotions or a perception of a third, more complex emotion.

So far, most studies on the perception of blends have used stimuli that are based on photographs [4]. However, from previous studies we know that there are some typical differences in how people perceive the basic emotions from natural and synthetic facial expressions, and therefore the perception of blends may also be different for natural and virtual faces. As for basic emotions, synthetic faces are typically poorer in communicating fear [9, 18] and disgust [9, 5], but on the other hand, they are often better than natural faces in communicating sadness [18, 5].

The main goal of this study is to investigate whether virtual agents can communicate complex emotions using blends of facial expressions of basic emotions. Due to differences in the perception of natural and synthetic faces, it is important to study the perception of blends specifically on synthetic faces. We use a physically-based facial model which is capable of producing all muscle actions required for the expressions of basic emotions, and combined muscle actions required for the blends.

We study all pairwise blends of basic expressions. The original basic expressions used to create a blend are called *parent expressions*, and the corresponding emotions are called *parent emotions*. The term *blend* refers to the combination of two parent expressions. Emotions that do not belong to the set of basic emotions are called *complex emotions*. The term *target emotion* is used in our analyses for an emotion that an expression is supposed to present (parents of a blend, or basic emotions in their original expressions).

This study consists of two experiments. The first experiment focuses on whether the used stimuli were perceived as pure basic emotions or combinations of two basic emotions. The second experiment focuses on whether blended facial expressions are perceived as expressing complex emotions.

2 VIRTUAL FACE MODEL

Various approaches can be adopted to model blends of facial expressions in virtual characters. Virtual faces themselves can be built in a variety of different ways, and several techniques can be used for constructing pure and blended facial expressions of basic emotions. The situation is made even more difficult by the fact that basic emotions and their blends can be expressed in several different ways even on a natural face.

To create the blends for this study, we used a facial animation model described in our previous work [13] (based on [20]). This facial model is physically based, and it has deformable skin and facial muscles. The facial tissue is implemented using a mass-string-damper model with two layers of cubical elements. The lower layer is attached to the bones which have been modeled beneath the tissue. Muscles are attached to the lower nodes in the top layer and to bone surface at the other end. The facial tissue has been modeled so that it is slightly asymmetric, similarly as in real human faces.

The animation model uses FACS (Facial Action Coding System) [7] as a control mechanism for facial expressions. FACS defines facial actions in terms of Action Units (AU). For example, AU12 is the action of the *Zygomaticus major* muscle pulling lip corners to a smile, and AU4 is the action of the *Corrugator supercilii* muscle making a frown. Our model includes facial examples of basic expressions that were created by selecting one prototypical AU combination defined in FACS for each emotion. The combinations used for the basic expressions are: joy 6+12 (with the addition of AU7), sadness 1+4+15, fear 1+2+4+5+20+26, anger 4+5+7+23, surprise 1+2+5+27 and disgust 10+17. The model also includes the possibility to blend any two facial expressions. This was implemented by adding the muscle activations of the parent expressions. This technique has advantages for creating blended expressions: muscle activations are anatomically correct, facial actions from both parent expressions remain present, and any pair of facial expressions can be blended easily.

3 EXPERIMENT 1: BASIC EMOTIONS IN BLENDS

The first experiment was designed to study the extent to which pure basic emotions are perceived in blends. Our secondary goal was to collect complex emotion words to be studied more thoroughly in Experiment 2. A further objective was to validate the basic expressions of our virtual face model.

29 volunteers were recruited via e-mails and social media. The sample consisted of 16 female and 13 male participants aged between 19 and 63 years ($M = 30.7, SD = 9.7$).

3.1 Methods

Using the model described above, we prepared 21 videos of expressions on a virtual face. They included the six basic emotions (anger, disgust, fear, joy, sadness and surprise) and all of their 15 pairwise blends. Neutral expression was not included, because dynamic but affectively neutral facial expressions would not

 ISBN 978-80-86943-41-1

have been analogous to our other stimuli. We recorded the transition from neutral to peak expression in a video clip with a duration of two seconds. The peak expression was reached in approximately one second.

As control stimuli, we used 21 corresponding videos produced by morphing basic emotion expressions posed by a human actor. These were selected from one actor (MO) in the Ekman and Friesen's Pictures of Facial Affect collection [8], which has become a standard database in this field. We used image morphing to blend basic expressions, which is a conventional method in facial expression studies (for example [4]). Photographs of each two basic expressions were blended with the ratio 50%–50% using the application MorphThing (http://www.MorphThing.com). Video sequences were created from the static expressions by morphing them with the neutral face using Sqirlz Morph software (http://www.xiberpix.com/SqirlzMorph.html). Similar approach has been used previously to create dynamic stimuli [17]. Only the face region was morphed, while the surrounding region was taken from the neutral face image.

Although forced-choice method is often used to study the recognition of basic emotions from facial expressions, this method is tied to a predefined list of emotion words and it makes the possibly incorrect assumption that a specific facial expression is only associated with one emotional state. In this study, we asked our participants to describe their perception of the facial expressions in more detail.

The evaluations were done using an online questionnaire. All 42 videos were evaluated one by one in random order. First, the participants rated each video on all six basic emotion dimensions using visual sliders on a scale ranging from no emotion (0) to extremely intense (100). This method enabled us to get detailed information not only about the recognized primary emotion, but also about the recognition of less intense secondary emotions. Second, to measure the recognition of complex emotions, the participants were asked to provide open responses to the question *"What other emotions do you see in the facial expression (if any)?"*

3.2 Analyses

To validate the modeled basic expressions, and to evaluate whether some of the blends are also perceived as *pure basic emotions*, we converted the six basic emotion ratings of each facial expression into a recognition score measuring whether one of them clearly dominates.

A facial expression can be thought to unambiguously display one *basic emotion* if people consistently give higher ratings to that emotion compared to all others. Thus, we defined the recognition score as the difference between the target emotion rating and the second highest rating (in case any other emotion received a higher rating than the target emotion, the score was negative). Formally, this recognition score can be defined as

$$RSpure_i(E) = R_i(E) - \max_{k \neq i}\{R_k(E)\}, \qquad (1)$$

where E is the facial expression, $R_k(E)$ is the rating for emotion k in the expression E, i is the targeted emotion, and k has six possible values: anger, disgust, fear, joy, sadness and surprise.

This recognition score is more accurate than a mean statistic for the targeted emotion, because it also takes into account how distinctive the target emotion was with respect to non-target emotions. A simple mean evaluation for a specific emotion can be high even though this emotion is considered a secondary emotion by most participants.

To measure whether a blend is recognized as a *mixed emotion*, we used a recognition score that has a positive value when both parent emotions receive higher ratings than any of the other emotions. This score is defined as

$$RSmix_{ij}(E) = \min\{R_i(E), R_j(E)\} - \max_{k \neq i; k \neq j}\{R_k(E)\}, \qquad (2)$$

where E is the facial expression, i and j are the targeted emotions with $R_i(E)$ and $R_j(E)$ their respective ratings, and $R_k(E)$ is the rating for a non-targeted emotion k. Again, the indices have six possible values: anger, disgust, fear, joy, sadness and surprise.

The score is positive only if the targeted emotions receive the highest and the second highest ratings among individual evaluations. Facial expressions meeting this strict requirement can be considered as unambiguous expressions of mixed emotions.

We used Wilcoxon signed-rank tests to determine whether the recognition scores were statistically

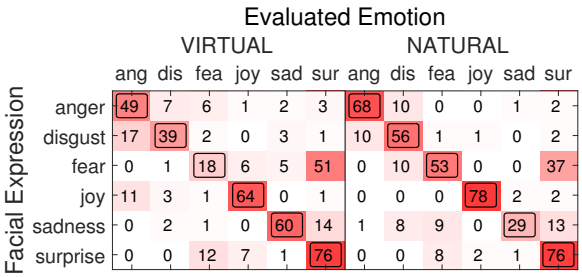


Figure 1: Perception of the expressions of basic emotions on virtual and natural faces presented as confusion matrices. Mean ratings of each emotion for each facial expression are presented as numbers and colour intensity (colour intensity is proportional to the number in the cell). Rectangles around numeric values indicate the targeted basic emotions.

ISBN 978-80-86943-41-1

emotion i	anger	disgust	fear	joy	sadness	surprise
$RSpure_i$ virtual	33.3***	18.5*	-33.3***	50.7***	45.6***	58.1***
$RSpure_i$ natural	56.7***	42.1***	11.4	74.7***	3.8	65.4***

Table 1: Virtual facial expressions of basic emotions at their emotional apex. Mean recognition scores $RSpure_i$ are listed for the basic expressions on virtual and natural faces. A positive score indicates that the expression was recognized correctly. Statistically significant scores are marked with asterisks.

emotion i emotion j	anger disgust	anger fear	anger joy	anger sadness	anger surprise	disgust fear	disgust joy
$RSmix_{ij}$ virtual	2.8	-38.4***	3.5	-7.3**	-16.5*	-42.1***	-19.0**
$RSpure_i$ virtual	**21.7****	-40.5***	1.7	-63.8***	-50.0***	-60.3***	-45.5***
$RSpure_j$ virtual	-31.4***	-19.7	-8.4	**54.7*****	17.1	-7.4	12.4
$RSmix_{ij}$ natural	9.7	-27.4***	-15.1**	-15.2**	-19.3**	-8.5	-3.3
$RSpure_i$ natural	-14.6	-5.7	**26.8****	14.9	-33.3***	**45.8*****	11.1
$RSpure_j$ natural	7.9	-37.9***	-47.9***	-37.7***	7.6	-59.0***	-22.8**

disgust sadness	disgust surprise	fear joy	fear sadness	fear surprise	joy sadness	joy surprise	sadness surprise
-0,1	-21,8**	-23,1**	-24,1**	8,1	-9,9*	**36,1*****	-25,3**
-63.8***	-45.9***	-40.6***	-26.5**	-45.8***	-0.4	-15.2*	-63.8***
61.5***	8.0	8.9	-11.4	**40.4*****	-22.2*	13.7	**23.1***
-5.5	-7.4*	-30.0***	-36.0***	**12.8***	-13.8**	2.7	-7.1
27.7***	**40.9*****	-29.1***	-13.3	-44.3***	0.9	-5.0	-31.8**
-35.1***	-49.9***	-17.1	-44.3***	**38.6****	-26.1***	-4.5	17.1

Table 2: The blends at their emotional apex displayed on the virtual face, and mean recognition scores $RSmix_{ij}$, $RSpure_i$ and $RSpure_j$ (statistically significant positive scores are in boldface).

different from zero. A nonparametric test was chosen because the recognition scores did not follow normal distribution. False-discovery rate correction at $\alpha = 0.05$ was applied to compensate for the multiple comparisons (102 comparisons: 6 for virtual basic expressions, 6 for natural basic expressions, 45 for virtual blends and 45 for natural blends). Statistical significance is indicated with the common asterisk notation * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ in Tables 1, 2 and 4.

3.3 Results

We first evaluated whether the basic emotion expressions were actually perceived as such. Expressions produced using the virtual face model were compared to expressions posed by a real human.

Figure 1 shows the mean emotion ratings for all *basic expressions* of virtual and natural faces. Visual inspection of this figure suggests that most targeted emotions are perceived as expected. The fearful virtual face was perceived as more surprised than fearful, and that also

ISBN 978-80-86943-41-1

the fearful natural face received high surprise ratings. With the natural face, the sad facial expression received remarkably low sadness ratings.

Recognition score results, visualized in Table 1, show that indeed all expressions except fear were recognized correctly from the virtual face, and all except fear and sadness were recognized correctly from the natural face. Thus we can be confident that our model reasonably well resembles a human face.

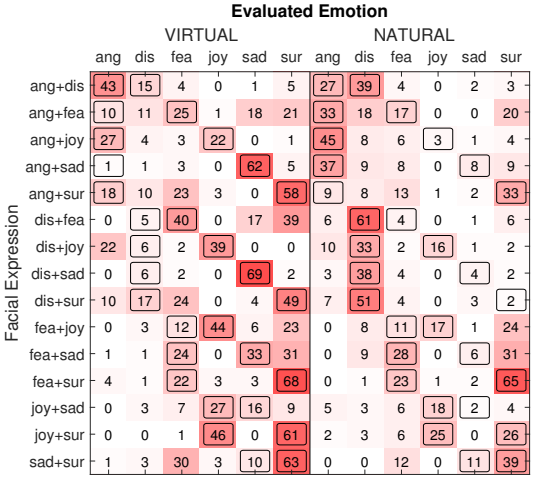


Figure 2: Perception of basic emotions in blends. The visualization is analogous to Figure 1.

The mean basic emotion ratings for all *virtual and natural blends* are presented in Figure 2. Visual inspection suggests that there are considerable differences between blends in how the ratings were distributed among basic emotions: Some of the expressions are recognized as one basic emotion, while others seem to express several emotions.

The recognition scores $RSmix_{ij}$ and $RSpure_i$ are collected in Table 2. They show that for the virtual face, only the blend of joy and surprise was recognized successfully as a mix of its parent emotions. Anger+disgust was perceived as anger, anger+sadness and disgust+sadness were both perceived as sadness, and fear+surprise and sadness+surprise were both perceived as surprise. As for the natural faces, only the blend of fear and surprise fulfilled the criteria of successful blend recognition, and even that blend was primarily perceived as surprise. Anger+joy was perceived as anger, whereas disgust+fear, disgust+sadness and disgust+surprise were all perceived as disgust.

As many of the blends were not perceived clearly as either mixtures of two basic emotions or instances of one basic emotion, the goal of our second experiment was to figure out whether some of the blends are perceived as expressions of more complex emotions. The open-ended part of our questionnaire offered a starting point: 17 participants out of 29 used the option of supplementing their answer using their own words. Altogether, additional descriptions were given 89 times. Experiment 2 was based on these answers.

4 EXPERIMENT 2: COMPLEX EMOTIONS IN BLENDS

To test whether some of the additional emotion words would describe perceptions of blends more accurately than the basic emotion words do, we paired these words with all facial expressions. The same videos of the virtual face that were used in Experiment 1 were used in this experiment also. The morphed natural faces were not used. Thus, there were 21 different facial expressions.

From all the additional words collected, we ignored references to non-genuine emotions (such as 'fake smile'). The remaining 17 words for complex emotions, together with the six basic emotions resulted in 23 emotion words altogether. When each facial expression was paired with each emotion word, we had $21 * 23 = 483$ expression-emotion pairs to study.

Using a three-stage procedure, we first narrowed down the list of expression-emotion pairs into those that appear more often than by chance, and then examined more carefully which expression-emotion pairs show evidence of complex emotion recognition.

This experiment was conducted using a crowdsourcing platform CrowdFlower (http://www.crowdflower.com/). The participants represented various backgrounds and were different in the different stages of the experiment.

4.1 Procedure and results

Stage 1 consisted of finding out which expression-emotion pairs are at least to some extent associated with each other. For each expression-emotion pair, the participants were asked *"Does this word describe the state of the person in this video?"* (yes or no). Ten evaluations were collected for each pair. All emotion words (with the corresponding expressions) that were mentioned six times or more were taken for further inspection in Stage 2. These words were: ambiguous, apologetic, disappointment, embarrassment, envy, malicious joy, revengeful, serious, shame, shock, and suspicious (words discarded at this stage were: cunning, concentration, despair, determination, interested and relief).

The purpose of Stage 2 was to further narrow down the list of expression-emotion pairs to only those with a strong association. All videos were shown to the participants with the question *"Which of these words best describes the emotion of the person in the video?"* The participants answered by making a forced-choice from one of the alternatives identified in Stage 1. It was also possible to select *"none of the above"*. 40 evaluations

ISBN 978-80-86943-41-1

Expression	Best descriptions of the expression	
	Targeted	Others
anger	anger	suspicious, revengeful, serious
disgust	disgust	envy
fear	fear	
joy	joy	malicious joy
sadness	sadness	
surprise	surprise	shock
ang+dis	anger	suspicious, revengeful
ang+fea	fear	disgust
ang+joy		malicious joy, revengeful
ang+sad	sadness	
ang+sur		shock
dis+fea	fear	disappointment
dis+joy	joy	malicious joy
dis+sad	sadness	
dis+sur		shock
fea+joy	joy	embarrassment
fea+sad	fear, sadness	disappointment
fea+sur		shock
joy+sad		ambiguous, apologetic, shame
joy+sur	surprise	
sad+sur		shock

Table 3: All emotion words that were selected as the best description for the corresponding expression more often than by chance. Targeted (basic) and other (complex) emotions in separate columns.

were collected for each facial expression. All emotion words that were mentioned more often than chance level as the best description for the facial expression were selected to the next list, presented in Table 3.

Stage 3 was conducted to identify blended facial expressions that are perceived as complex emotions. Candidates for this are the words in the bottom-right cell of Table 3, column "Others" for blended facial expressions. In total, 16 complex emotion terms for 12 blended facial expressions were considered at this stage (we also included as 'complex' the basic emotion word disgust for the blend anger+fear, because it is neither of its parent emotions).

For each of the selected pairs between complex emotions and blended facial expressions, participants were asked to answer the question *"Which of these words best describes the emotion of the person in the video?"* with three possible choices: the complex emotion or either of its parent emotions. The same question was answered separately for three videos: the blended facial expression and both of the parent expressions. Each evaluation was conducted 120 times, resulting in nine frequency scores for each blend-emotion pair: three videos times three words.

Based on these scores we defined two indexes called *association* and *distinctiveness*. Association is positive if a complex emotion word describes the blend more accurately than either one of the parent emotion words. Distinctiveness, on the other hand, is positive if a complex emotion is associated specifically with the blend in contrast to the parent expressions. Formally we define them as

$$Ass(c,B) = S_c(B) - \max\{S_1(B), S_2(B)\}, \quad (3)$$

$$Dis(c,B) = S_c(B) - \max\{S_c(P_1), S_c(P_2)\}, \quad (4)$$

where c is a complex emotion, B a blend formed from parent expressions P_i (i=1..2), and $S_x(E)$ is the score of an emotion x (complex or parent) for the video of expression E.

Pearson's chi-square test with Yates' correction for continuity was used to determine whether association and distinctiveness values are different from zero with a statistical significance.

The results are collected in Table 4. The expression-emotion pairs are divided into three groups based on the strength of evidence they provide for the hypothesis that the complex emotion word unambiguously describes the blend (strong if both association and distinctiveness are positive, and weak if distinctiveness only is positive, and no evidence otherwise). For majority of the positive findings, parent emotions appear to be joy, sad, anger or surprise.

Expression-emotion pair	Association	Distinctiveness
joy+sad = apologetic	53***	73***
joy+sad = ambiguous	49***	71***
joy+sad = shame	43***	68***
joy+ang = malicious joy	58***	65***
joy+dis = malicious joy	59***	63***
joy+ang = revengeful	28***	54***
sur+sad = shock	28***	45***
sur+ang = shock	17*	44***
joy+fea = embarrassment	13	38***
sur+dis = shock	4	32***
sur+fea = shock	-8	28***
fea+ang = disgust	-45***	27***
ang+dis = revengeful	-70***	10
ang+dis = suspicious	-54***	7
fea+dis = disappointment	-77***	-4
sad+fea = disappointment	-49***	-8

Table 4: Association and distinctiveness values for all included expression-emotion pairs.

5 DISCUSSION

The present results show that most basic emotions were recognized very well from our virtual character. Although fear was incorrectly recognized as surprise, this confusion was also present in the natural face. Even though we did not explicitly compare recognition scores for the virtual and natural face, we note that the intensities tended to be lower for the virtual face. This observation is consistent with several previous facial animation studies [11, 9, 19, 18].

Only one virtual blend, joy+surprise was perceived as a mixture of its parent emotions. The natural blend fear+surprise was also recognized as a mixture of these two, but the perception of surprise was dominating. Also, it is noteworthy that the pure basic expression

 ISBN 978-80-86943-41-1

of fear was perceived as surprise both in our virtual and natural faces. The present scoring method was relatively strict, which may partly explain why no other blends were reliably recognized. However, the results show that it is not reasonable to assume that both parent emotions could be recognized from blends in general.

Our analysis revealed that five virtual blends and five natural blends were perceived as expressions of one of the parent emotions (bolded scores in Table 2). In the case of virtual fear+surprise blend, this result is trivial, since the facial expression of fear was also perceived as surprise. The other blends that were perceived as a parent emotion could be seen as partial evidence supporting the categorical emotion view. According to that view, when an expression gradually moves from anger to disgust (for example), it is perceived as pure anger until after a certain point it is perceived as pure disgust. Although congruent with our observations, only a minority of the blends were perceived as parent emotions, and thus we can't expect that this would generally happen.

The reason why some virtual blends were perceived as their parent emotion may be related to the blending method, which added together all muscle activities from both parent expressions. Some of the basic expressions include much greater and/or more visible movements than others, and thus in a blend the subtle movements may be overshadowed by the more prominent movements.

The blends that were strongly or weakly perceived as complex emotions (the two upper sections of Table 4) can be divided into two groups based on which parent expressions they consist of. The first group is joy blended with a negative emotion. These blends produce a variety of complex emotions which (with the exception of malicious joy) seem to be unique for each blend. However, a single blend can be described with several different emotion words. This is in accordance with the view that interpretation depends on context.

The second group is surprise blended with a negative emotion. All of these blends can, to some extent, be described with the word shock, which is believable considering the emotional content. This result may be somewhat questionable, however, because on Experiment 2 Stage 2, the basic expression of surprise was also often described as shock.

5.1 Limitations

This study was conducted using a single virtual face model and a single blending algorithm. The blends were created using only one facial expression of each basic emotion category. Although the used expressions of basic emotions were found relatively recognizable in comparison to natural facial expressions from a standard collection, they are not perfect, and specifically the expression of fear was poorly recognized. A wider variety of basic emotions could be used to create blends, and the faces could represent different individuals. Moreover, other animation methods for creating blends besides our additive method could be tested. Future studies might also consider whether different blending proportions of two expressions would produce different results.

Our facial model is crude compared to the highly photorealistic models used in movie industry. However, its visual fidelity is comparable to that of contemporary virtual agents used in interactive virtual reality and games. More advanced modeling and rendering may add details, such as wrinkles, that may cause new perceptual effects and different results.

Our stimuli were dynamic, but the brief motion from neutral to peak expression is still somewhat artificial. In real conversational situations facial expressions change continuously and follow each other. Some emotion blends may be expressed with two consecutive expressions, and blends are likely to momentarily occur when the emotional state changes. These kinds of temporal aspects are important in developing believable animated characters and virtual agents, and therefore future research should address also this issue.

6 CONCLUSION

As animated virtual characters become more humanlike, expectations towards their facial behavior increases. To be able to create believable facial expressions of emotions that imitate expressions of real humans, we need more understanding about how different facial expressions of virtual characters are perceived.

Our results demonstrate that people are often not able to correctly recognize the two basic emotions in a blend of facial expressions, but instead, some blends produce a perception of another, complex emotion. The blend of surprise with any negative emotion is often labeled as shock. On the other hand, blends with opposite valence (joy combined with a negative emotion) can be described with various complex emotion words. In real applications, the interpretation of these facial expressions would probably depend on context.

The results indicate that blended facial expressions of basic emotions can be used to increase the emotional expressiveness of virtual agents. To communicate more complex emotional states in addition to the basic emotions, it is important to blend not only facial expressions of emotions that are close to each other in a conceptual emotional space, such as anger and disgust, but also facial expressions that represent opposite emotional states, such as joy and sadness.

 ISBN 978-80-86943-41-1

To our knowledge, this is the first study to systematically search for perceptions of complex emotions in pairwise blends of basic expressions. Its main contribution is to outline methodology and lay hypotheses for further research, while the detailed results and the scores used in the analysis may need revised studies with different facial models.

7 ACKNOWLEDGMENTS

This study was supported by the Academy of Finland, Doctoral Program in User-Centered Information Technology (UCIT), and by the H2020-MSCA-IF-2015 grant (no. 703493) to Jari Kätsyri.

8 REFERENCES

[1] Junghyun Ahn, Stephane Gobron, Daniel Thalmann, and Ronan Boulic. Asymmetric facial expressions: revealing richer emotions for embodied conversational agents. *Computer Animation and Virtual Worlds*, 24(6):539–551, 2013.

[2] Irene Albrecht, Marc Schröder, Jörg Haber, and Hans-Peter Seidel. Mixed feelings: expression of non-basic emotions in a muscle-based talking head. *Virtual Reality*, 8(4):201–212, 2005.

[3] Ali Arya, Steve DiPaola, and Avi Parush. Perceptually valid facial expressions for character-based application s. *International Journal of Computer Games Technology*, 2009, 2009.

[4] Andrew J Calder, Duncan Rowland, Andrew W Young, Ian Nimmo-Smith, Jill Keane, and David I Perrett. Caricaturing facial expressions. *Cognition*, 76(2):105–146, 2000.

[5] Miriam Dyck, Maren Winbeck, Susanne Leiberg, Yuhan Chen, Rurben C. Gur, and Klaus Mathiak. Recognition profile of emotions in natural and virtual faces. *PLoS ONE*, 3(11):e3628, 11 2008.

[6] P Ekman, WV Friesen, and P Ellsworth. What emotion categories or dimensions can observers judge from facial behaviour? in, p. ekman. *Emotion in the Human Face*, 1982.

[7] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System: Investigator's Guide*. A Human Face, 666 Malibu Drive, Salt Lake City UT 84107, USA, 2002.

[8] Paul Ekman, Wallace V Friesen, and Consulting Psychologists Press. *Pictures of facial affect*. consulting psychologists press, 1975.

[9] Marc Fabri, David Moore, and Dave Hobbs. Mediating the expression of emotion in educational collaborative virtual environments: an experimental study. *Virtual Reality*, 7(2):66–81, 2004.

[10] Scott H. Hemenover and Ulrich Schimmack. That's disgusting! ..., but very amusing: Mixed feelings of amusement and disgust. *Cognition & Emotion*, 21(5):1102–1113, 2007.

[11] Jari Kätsyri, Vasily Klucharev, Michael Frydrych, and Mikko Sams. Identification of synthetic and natural emotional facial expressions. In *AVSP 2003-International Conference on Audio-Visual Speech Processing*, 2003.

[12] Jeff T Larsen and A Peter McGraw. The case for mixed emotions. *Social and Personality Psychology Compass*, 8(6):263–274, 2014.

[13] Meeri Mäkäräinen and Tapio Takala. An approach for creating and blending synthetic facial expressions of emotion. In *IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pages 243–249, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] Jean-Claude Martin, Radoslaw Niewiadomski, Laurence Devillers, Stephanie Buisine, and Catherine Pelachaud. Multimodal complex emotions: Gesture expressivity and blended faci al expressions. *International Journal of Humanoid Robotics*, 3(3):269–291, 2006.

[15] Radoslaw Niewiadomski, Magalie Ochs, and Catherine Pelachaud. Expressions of empathy in ecas. In *Intelligent virtual agents*, pages 37–44. Springer, 2008.

[16] Catherine Pelachaud, Norman I. Badler, and Mark Steedman. Generating facial expressions for speech. *Cognitive Science*, 20(1):1–46, 1996.

[17] Wataru Sato, Takanori Kochiyama, Sakiko Yoshikawa, Eiichi Naito, and Michikazu Matsumura. Enhanced neural activity in response to dynamic facial expressions of emotion: an fmri study. *Cognitive Brain Research*, 20(1):81–91, 2004.

[18] Angela Tinwell, Mark Grimshaw, Debbie Abdel Nabi, and Andrew Williams. Facial expression of emotion and perception of the uncanny valley in virtual characters. *Computers in Human Behavior*, 27(2):741 – 749, 2011.

[19] T. Wehrle, S. Kaiser, S. Schmidt, and K. R. Scherer. Studying the dynamics of emotional expression using synthesized facial muscle movements. *Journal of Personality and Social Psychology*, 78:105–119, 2000.

[20] Yu Zhang, Edmond C. Prakash, and Eric Sung. Efficient modeling of an anatomy-based face and fast 3d facial expression synthesis. *Computer Graphics Forum*, 22(2):159–170, 2003.

 ISBN 978-80-86943-41-1

Educational Virtual Environment Based on Oculus Rift and Leap Motion Devices

Matea Zilak
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia
matea.zilak@fer.hr

Zeljka Car
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia
zeljka.car@fer.hr

Gordan Jezic
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia
gordan.jezic@fer.hr

ABSTRACT

Virtual reality (VR) technology offers numerous benefits in different application areas, especially in education. VR brings new approaches to learning that can make the education process more attractive, while at the same time learners can develop creativity and innovativeness. Despite the possible benefits that VR can offer, the use of VR is still not widespread for educational purposes. Furthermore, the potential of VR for assistive technologies in the Augmentative and Alternative Communication (AAC) domain is recognized but has not been fully exploited. In this paper, development of an elementary mathematical virtual classroom prototype based on Oculus Rift and Leap Motion devices is described. The learning concept used in the prototype was taken from the state-of-the-art AAC application for mobile devices that introduces children with the concept of quantity which is one of the preconditions for adopting the concept of number. To analyze user's satisfaction with the application and acceptance of a new technology in general, user evaluation of the developed prototype was conducted. In general, a positive feedback from users without disabilities suggests that it makes sense to combine VR elements with education as well as AAC technologies. Contributing factors, such as the level of immersion in VR environments, unnatural behavior of virtual hands, and the level of familiarity with the VR technology, are identified as some of the most important aspects that need to be considered in the follow-up studies concerning users with disabilities (i.e. children with complex communication needs).

Keywords

Human Computer Interaction; Virtual Reality; Educational; Oculus Rift; Leap Motion; User Evaluation; AAC

1. INTRODUCTION

Within the last few years Virtual Reality (VR) technology has experienced growth of its popularity which had an impact on development of VR applications for practical use in areas other than entertainment and gaming, such as education. Much research has already been conducted on the application of VR in education which revealed numerous benefits that VR offers in this area [Pan10]. Traditional teaching and learning methods require little or no interaction with a student which makes them very static and, consequently, student's attention cannot be kept for a long time [Ray16]. On the other hand, VR requires interaction and encourages active participation rather than passivity, which has an impact on increased level of motivation of students towards learning [Pan10].

Youngblut in [You98] presented unique capabilities of VR technology, such as the ability to visualize abstract concepts, to observe events at atomic or planetary scales, and to provide teaching in virtual environments that are impossible to visualize in physical classroom due to different safety, distance or time factors. Because of many examples of abstract problems that it provides, mathematics is an area in which VR can help learners to visualize abstract mathematical facts and understand problems that require, for example, spatial skills which are sometimes hard to understand for students [Kau09]. An extensive study on the educational applications of VR conducted by authors in [Mik11] revealed that majority of empirical studies they reviewed refer to science and mathematics topics which suggests that these areas offer a number of challenges that can be addressed with the use of VR technology.

Even though VR systems are now more acceptable and affordable than before, there are still people who have never experienced it and, therefore, never used it

for educational purposes. Other types of educational technologies are used instead, such as smart boards, long-distance learning and mobile learning. Since the use of virtual reality is still not widespread, we have been motivated to propose a prototype of elementary mathematical virtual classroom with which a user is able to interact. In particular, human computer interaction in this prototype is based on the mechanism by which the user can interact in an intuitive way, with his/her hands. The objective behind the development of this kind of VR application is to analyze a new method of learning, based on modern technology such as virtual reality, and its acceptance.

The authors in [Dun12] described benefits that virtual worlds can have in teaching and learning, as they can provide socialization, entertainment and a laboratory for collaborative work. However, they emphasized that the use of virtual worlds for education should not disadvantage particular social, minority or disabled groups. According to [Lig07], technologies such as VR educational and play environments are an example of assistive technologies that may offer children with complex communication needs (CCN) the means to interact positively with nondisabled peers (e.g. the classmates, neighbors, potential friends of children with CCN) on an equal footing. That way, children with disabilities can overcome significant attitudinal barriers they confront with in society and can increase their self-esteem and self-empowerment [Lig07]. Augmentative and Alternative Communication (AAC) technologies support the communication process of persons with CCN. As a part of a research group which conducts intensive research in the AAC domain over nine years, especially within the EU funded ICT-AAC project [Ict13], we have developed over 30 AAC applications for most popular platforms, such as Android, iOS and Web. The idea behind the development of the applications was to make the learning process and communication as attractive as possible to encourage users to use the applications. For our AAC applications to be accessible and highly usable, in development process of every AAC application, as described in [Bab15], we cooperated with experts from different fields, such as rehabilitation and education and graphic design. The research in the AAC field is also analyzed from a technical point of view where technology capabilities regarding Machine-to-Machine communication are investigated. To properly interpret information about the user and his environment, communication and interoperability of different systems are necessary, so *machine social networks* [Pti16] impose as a possible solution for AAC systems to adapt as much as possible to the user. Example of a system like this can be a VR environment which provides a context to support social interaction for children with CCN. Since the potential of VR technology in AAC domain has not quite been exploited [Lig07], we have been additionally motivated to explore possible benefits that VR can bring in the AAC domain. Although our AAC applications, which are mostly tablet-based, have a stable and growing user base [Bab15], we are intrigued to cope with a challenge in developing VR-based AAC applications that can make a difference for children with CCN.

Considering previously written and the fact that there is a wide range of potential uses of VR in education, especially in mathematical domains for younger students as well as university level students [Kau09], a VR application that provides experience of learning basic mathematical concepts for younger children can be considered as a convenient option for the first step in the process of introducing VR for educational purposes (to a wider mass). That is why the prototype we proposed was developed based on one of the AAC applications developed within the ICT-AAC project. In the prototype we developed, the Oculus Rift headset is used for immersion into the virtual world while the Leap Motion sensor device is used for hand tracking. To analyze user satisfaction with usage of the VR application we developed and acceptance of the use of new technology in general, user evaluation is conducted and documented within this paper.

The paper is organized as follows: Section II details related work, Section III describes the AAC application on which development of the prototype was based, Section IV brings the description of the system architecture, implementation and functionalities, the process and the results of user evaluation are described in the Section V, and Section VI concludes the paper and presents a few ideas for future work.

2. RELATED WORK

Many VR environments for various educational purposes can be found. In [You98] Youngblut described many solutions developed for various educational purposes. One of the notable solutions is MaxwellWorld, an example of application developed as a research tool that provides a fully immersive and multisensory interface. Students interact with the virtual environment using virtual hands and menus (a Polhemus 3-Ball device [Ded96] is used for selection of menu item). Evaluation of the use of MaxwellWorld resulted in finding some important characteristics that aided learning, such as 3D representations, the interactivity, the ability to navigate to multiple perspectives, and the use of color.

 ISBN 978-80-86943-41-1

In addition to that, when compared to the EM Field[1] computer-based simulator, MaxwellWorld is rated as easier to understand (due to better representations), but as harder to use (due to troubles while using the 3-Ball and virtual hand) [Ded96]. As students differ in their interaction styles, interaction with virtual environment should be intuitive, understandable and well-known. User interface with those characteristics is called a Natural User Interface (NUI) and includes the use of devices that enable such interaction, such as Leap Motion [Lin16].

Educational virtual environment based on visualization of procedures and abstract concepts positively influences learning, especially if the user communicates with the virtual world in the form of an interactive game. That way, the user learns efficiently, but in an entertaining way, as the authors described in [Gri16] to be the case with the innovative educational environment for learning search algorithms, a topic which is often considered challenging for students to master.

Example of a virtual environment that uses key elements of successful computer games and emotionally appealing graphics is SMILE[2] (Science and Math in an Immersive Learning Environment), one of the first bilingual immersive virtual learning environments for deaf and hearing students. The user interacts with virtual characters using the sign language by learning mathematical concepts and mathematical terminology of the American Sign Language (ASL). Formative evaluation of the game showed that the children perceived the game as more fun and easier to use and slightly more challenging than expected [Ada07].

Not many VR applications where the Oculus Rift and the Leap Motion are exclusively used can be found in literature, especially for educational purposes, but some of them can be identified as notable uses. The authors in [Lin16] described a VR system where the Oculus Rift and the Leap Motion are used. The purpose of a system they developed is to facilitate the selection of scientific articles which can be useful to researchers in their work. They proposed a new interface in which user interacts with his/her own hands and voice to enable natural interaction. Authors investigated different interaction techniques for immersive virtual environments including selection techniques and concluded that some of them are less intuitive than others, e.g. manipulation by gestures is less intuitive than direct manipulation in which selection of objects by virtual hand is identical as it is in the physical world [Lin16].

VR application the authors described in [Ala17] is developed to solve some of the educational problems which are still present among students, such as the lack of student's attention and difficulties to visualize what is being taught. Furthermore, virtual environment in which different experiments can be done is provided to avoid injuries that might occur in real environment because of an improperly conducted experiment. Also, this application enables easier performance of experiments to the disabled students because the use of VR headset and motion controller helps them to avoid movement struggles they usually have [Ala17].

After the literature survey on VR technology application areas, we specified the following for the elementary mathematical virtual classroom prototype:

- for the prototype to be easily understandable to children, natural interaction and intuitive selection technique will be achieved with the use of Leap Motion device, and
- for the prototype to be appealing to children, learning will be realized through an interactive game augmented with appropriate graphics.

3. ROLE-MODEL APPLICATION

As mentioned earlier, development of the prototype was based on one of the AAC applications developed within the ICT-AAC project. The application chosen to be the role-model application in this work is the ICT-AAC Domino counter[3] mobile application. It is an application for mobile devices (smartphones, tablets) with Android or iOS operating systems. This application provides the children with developmental disabilities early experiences with definition of quantity and numbers in an easy and attractive way, enriched with appropriate images and sound recordings. Knowledge of quantity is one of the prerequisites for adopting the concept of number and basis for future calculation. In addition to this, use of ICT-AAC Domino counter goes from the use in family environment and/or pre-school institutions to use in the initial stage of math teaching in elementary schools. Although ICT-AAC Domino counter is primarily intendent for children with disabilities, the application can also be used by young children of typical development where there is no need for additional professional support [Ict14]. For these reasons, ICT-AAC Domino counter is considered as a suitable application to be the role-model when

[1] EM Field by D. Trowbridge and B. Sherwood, http://www.physics.umd.edu/rgroups/ripe/software/emfield.html

[2] SMILE, http://hpcg.purdue.edu/idealab/smile/about.html

[3] ICT-AAC Domino counter on Google Play Store, https://play.google.com/store/apps/details?id=hr.fer.ztel.ictaac.domino_brojalica

 ISBN 978-80-86943-41-1

developing the prototype of elementary mathematical virtual classroom.

Learning Concept of ICT-AAC Domino Counter Application

The application helps users to learn about the quantity by using the so-called *domino principle* - by linking a certain number of symbols with a corresponding number of dots on domino tiles. Within the application, users can learn about quantity by counting symbols or by recognizing the given number. At the beginning, the user is offered to choose between four possible game levels (learning to three, five, seven or ten) and, after that, between two possible game modes (playing with numbers or symbols). Depending on which game mode the user has chosen, tasks in the game are displayed by numbers or by symbols which the user needs to associate with appropriate answers displayed in the form of domino tiles. Home screen of the ICT-AAC Domino counter application is shown in Figure 1. The application has settings in which it is possible to choose between different options to customize the interface (e.g. choose between growing and mixed order of the domino tiles, select the number of tasks displaying in one round as well as the number of answers (domino tiles)).

Figure 1 Home screen of the ICT-AAC Domino counter application

Serious Game Design Elements in ICT-AAC Domino Counter Application

In addition to being intended for younger children and enhancing the early math literacy skills required for later understanding of basic calculation, the ICT-AAC Domino counter application already has several game design elements that enhance the efficiency of educational tools and which can be utilized when implementing the prototype of an elementary mathematical virtual classroom. The importance of game design elements is explained by authors in [Ada07], who defined several game design elements that promote motivation of children to play the game again, enjoyment, and, therefore, learning. These elements are: a clearly defined background story and an overall structure of the game that gives meaning to all the activities of the game, the overall goal of the game, virtual world represented in a visual style that is appealing to the target age group (for children it is cartoon-like), multiple levels with variable difficulty, rewards associated with advancement, tips instead of answers. Besides the game design elements, design of interaction also has a role in encouraging user to continue playing – increased possibility of interaction with virtual world has a positive influence. That being said, some of the features of the ICT-AAC Domino counter are as follows:

- a graphical user interface presented in a style that is appealing to the target age group,
- multiple levels with variable difficulty,
- prominent progress through the game and
- feedback on the correctly/incorrectly answered task in visual and acoustic form.

4. MATHEMATICAL VIRTUAL CLASSROOM PROTOTYPE

System Architecture

An overview of the system architecture of the *Mathematical Virtual Classroom* is shown in Figure 2. One can notice that there is a two-way interaction between a user and the system. Firstly, the user's head and hand movements are monitored by two input units of the system: i) *Leap Motion* (i.e. sensor that tracks hands movements) and ii) *Oculus camera* (i.e. camera that tracks user's headset movement). Input data, such as head tracking data (e.g. headset's orientation and position) and hand tracking data (e.g. palm's and fingers' position and direction), are then transferred to the computer which runs the VR application developed within the Unity3D game engine. The retrieved data is then processed in real time and the appropriate simulation of the virtual environment and user's virtual hands is generated. Rendered image of the 3D world is then displayed on two output units at the same time: i) the *Oculus Rift head-mounted display* and ii) a *diagnostic screen*. Furthermore, audio output from the application is sent to the speaker which is responsible for giving the user audio feedback depending on user's actions during the game (e.g. positive audio feedback for a correct answer).

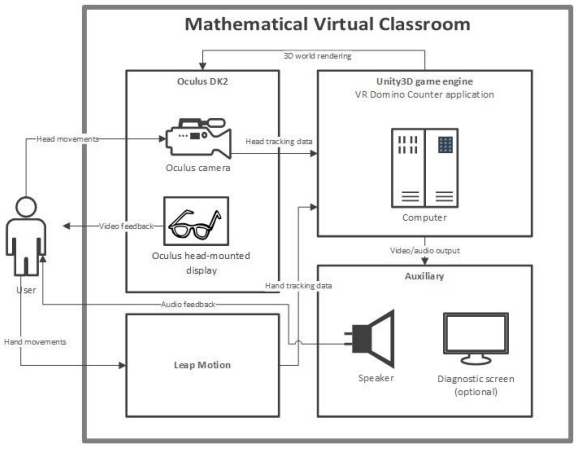


Figure 2 The system architecture of the *Mathematical Virtual Classroom*

ISBN 978-80-86943-41-1

Assets Used in the Mathematical Virtual Classroom

For development of the mathematical virtual classroom prototype, the Unity game engine (version 5.5.2f1) was used. To enable immersion into and interaction with virtual environment, the Oculus Rift and Leap Motion modules are integrated with Unity software. Unity has built-in VR support for the Oculus Rift and the Leap Motion. The Oculus offers optional utilities including different scripts, prefabs, and other resources to assist with development. For development of the mathematical virtual classroom based on the ICT-AAC Domino counter application, resources from the Oculus Utilities 1.3.2 package were used. To develop VR application using the Leap Motion, it was needed to retrieve the Leap Motion Orion Beta software for development. In our project the Leap Developer Kit 3.2.0 was used. Additionally, to access the classes and functions offered by the Leap Motion Application Program Interface (API), Unity needs to include the Unity Core Assets basic package. In our project the Leap Motion Orion Beta 4.2.0 package was used. Extensions like the Hands Module 2.1.2 and the UI Input Module 1.2.1 are used to facilitate development of user interface and design of hands models.

Besides assets mentioned above, other assets from the official Unity asset store and online stores of 3D models were used to display terrain, the background of the scene and various objects in the virtual environment such as trees, wooden panel on which different UI elements are shown, wooden table on which domino tiles as answers appear, domino tile models etc. Figure 3 shows what user sees in the virtual environment when he/she is in the middle of the terrain with extended hands in front of him.

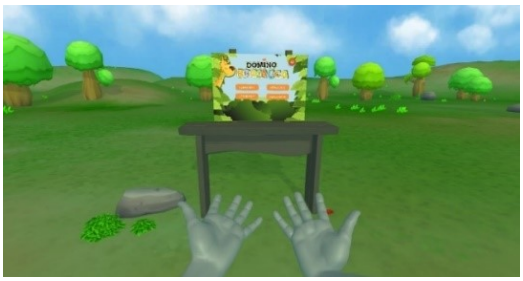


Figure 3 User's view from the middle of the terrain

Basic Functionalities

Because prototype development was based on the ICT-AAC Domino counter application, it was necessary to realize most of the functionalities that the ICT-AAC Domino counter has. The flow of the VR application is the same as the flow of the Domino counter application in general – at the beginning the user selects one of four possible game levels and one of two possible game modes followed by displaying problematic tasks in the form of numbers or symbols that user needs to associate with appropriate domino tiles. It is also possible for a user to change some settings, e.g. change the number of tasks or answers displaying in one round and change the order of domino tiles displaying (growing or mixed order).

In the ICT-AAC Domino counter application, the user interacts with the system by using simple gestures such as pushing buttons displayed on the touchscreen, while in the mathematical virtual classroom the user is interacting with virtual objects with his/her (virtual) hands. The interaction technique used includes selection of objects by virtual hand – for selection of game levels, game modes and problematic tasks displayed on the panel, appropriate gesture such as pushing buttons is used. In order to link certain task with appropriate answer, user needs to touch the 3D domino tile model displayed on the table.

Figure 4 shows what user sees when choosing between playing with numbers or symbols. An example of the task displayed by numbers is shown in Figure 5 while an example of the task displayed by symbols is shown in Figure 6. An example of a game moment when the task is answered incorrectly is shown in Figure 7 while an example of correctly answered task is shown in Figure 8.

Figure 4 User's view when choosing the game

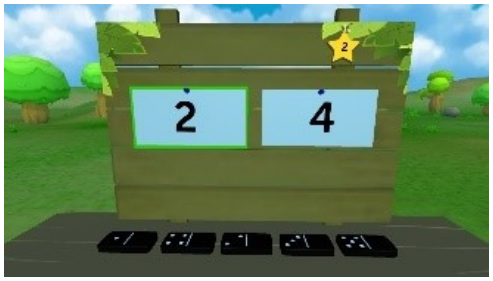


Figure 5 Task example displayed by numbers

Figure 6 Task example displayed by symbols

ISBN 978-80-86943-41-1

Figure 7 Example of a task answered incorrectly

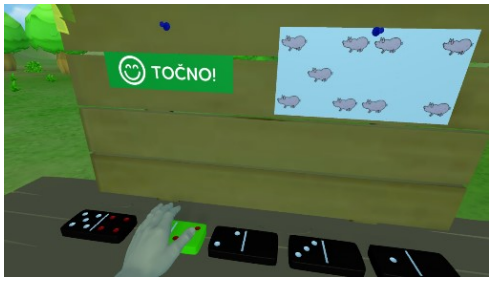


Figure 8 Example of a task answered correctly

5. MATHEMATICAL VIRTUAL CLASSROOM USER EVALUATION

To conduct user evaluation of the developed mathematical virtual classroom prototype, subjective and objective measures were specified. Objective measures related to the game play time and the number of incorrect answers were implemented in the software application itself while subjective measures of user satisfaction were collected through anonymous questionnaires after the application was used. The prototype was evaluated on a sample of 30 students of different gender and age. The login feature was also added to be able to distinguish measures for each user.

Setup for the Experiment

To use the elementary mathematical virtual classroom properly, it is necessary to do next: the VR application needs to be run on the computer while the user needs to put the Oculus Rift with the Leap Motion attached on his/her head. In addition, it is necessary to ensure that the camera is placed in the appropriate position to track the user's head position (in this case camera should be placed on the top and the center of the monitor). Also, the user should be sitting during interaction to reduce the possibility for motion sickness (otherwise, it is possible for user to play a game in a standing position as well as to explore the environment walking around on a distance that is permitted by the cables). By launching an application, the user is immersed into the virtual environment that is displayed on the head-mounted display. At the beginning, each participant entered his/her username using the virtual keyboard and after that, the user was introduced with how the application works by selecting one of three game levels: learning to 3, 5 or 7. After successfully completing the selected game, the participant activated the measurement of objective measures by selecting the game level "Learning to 10". Figure 9 shows a user using the application while wearing the Oculus Rift with attached Leap Motion.

Figure 9 User interacts with virtual world objects

Results of Objective Measures

Since the prerequisite for successful use of the VR application is that the user is familiar with the concept of using virtual hands to interact in the virtual environment, for the purposes of user evaluation, the Expert is defined as a user who is experienced in using the VR application. Table 1 shows the comparison of the average playing time of the participants and the playing time of the Expert. As it can be seen from Table 1, the participants' average playing time needed for successful completion of the game is 2.5 times greater than the playing time of the Expert.

User	Participant	Expert
Average playing time [*s*]	58.02	23.19

Table 1 Comparison of participants' average playing time and playing time of the Expert

None of the participants had a shorter game play time than the Expert which was expected. Also, there is a noticeable difference between the results of each participant. The shortest playing time of the participants is 26.88 seconds, which makes it only about 3 seconds slower than the time of the Expert, while the longest playing time is 101.23 seconds. Total of 11 participants out of 30 (about 36%) had at least one incorrect answer during the game. It was expected that the participants who had a greater number of incorrect answers will have a longer playing time, but as it can be seen from the graph shown in Figure 10, there is no significant correlation between the duration of time participant spent in-game and number of incorrect answers. These results indicate that the performance of the game depends on how the individual has accepted immersion into the virtual world and the way of interacting with the virtual world, and how the individual felt while using this type of technology.

ISBN 978-80-86943-41-1

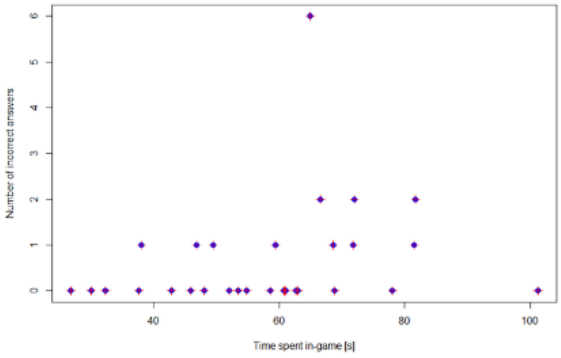


Figure 10 Correlation between time spent in-game and number of incorrect answers

Results of Subjective Measures

To collect subjective measures of satisfaction, each participant approached the questionnaire after he/she used the application. The ratings (marks) given for every question fall within a *Likert scale*, where the mark "1" is interpreted as "strongly disagree" and "5" is interpreted as "strongly agree". Table 2 shows seven questions used in the user satisfaction questionnaire and average mark calculated for each question. On the fifth question participant answered only if he/she had at least one incorrect answer.

Question Number	Question	Avg. Mark
1	The application was easy to use.	4.93
2	I was feeling comfortable while using the application.	4.8
3	I am satisfied with the speed of the application.	4.93
4	Interaction with user interface elements (e.g. buttons, domino tiles) was intuitive.	4.9
5	Incorrect answers were the result of intentionally choosing the wrong answer (rather than unnatural behavior of virtual hands).	4
6	I like the principle of immersion in the virtual world and the possibility of interaction with virtual hands.	4.87
7	Overall, I am satisfied with the application.	5

Table 2 Questions from the user satisfaction questionnaire and their average mark

The participants answered the last question unanimously with mark 5, meaning they are generally satisfied with the application. Because participants had little or no previous experience with VR, their satisfaction probably stems from the fact that they have not had the opportunity to experience VR in this way, where the interaction with the user interface and game objects is done in a natural and intuitive way (with virtual hands following the movements of their hands). This can also be seen from very high average marks regarding application ease of use (4.93), speed (4.93) and intuitive interaction with UI elements (4.9). A little lower average mark, but still very high, is related to the subjective feelings of the user, that is, the feeling of comfort during the use of the application (4.8) and the feeling of liking the principle of immersion into the virtual world and the possibility of interaction with virtual hands (4.87). These slightly lower marks are probably because the use of VR is still not widespread, so users are still not used to the feeling of full immersion into the virtual environment.

The lowest average mark (4.00) is calculated for the question regarding tasks that were answered incorrectly. Participants were supposed to answer whether inaccurate answer(s) were the result of intentional selection of the answer that was wrong or unnatural behavior of virtual hands. This question was asked because sometimes it may happen that the information obtained from the Leap Motion sensors is wrongly interpreted. This is most commonly occurring when the application is started or when some other object in the background engages in an area where the position of the hand is tracked. If this happened at the time a user needed to answer, it could be the reason why the user unintentionally answered incorrectly. Of all the participants who answered the fifth question only three of them "blamed" the application, i.e. the unnatural behavior of the virtual hands, which means that the wrong interpretation positioning of the hand position does not occur frequently.

The graph in Figure 11 shows average mark calculated from marks that participants gave for each question. Participants who gave average marks lower than 4.5 are the ones who gave mark 1 for the fifth question, but duration of their game play was shorter than average play time. Two participants who gave marks lower than the average (<4.7) but greater than 4.5 had longer playing time than average. These results (lower marks from participants who experienced unnatural behavior of virtual hands and who had longer game play time than average) indicate a certain correlation between the results from objective and subjective measures. The rest of the participants gave average marks greater than 4.7, and their playing time varied (shorter and longer than average). Interesting fact to note is that 12 participants out of the 30 (40%) gave the average mark equal to 5, and that group of participants includes both the person with the shortest and the person with the longest playing time. From this we can conclude that the reason the people who gave great marks played the game for a long time was not due to problems during the game but because they liked the feeling of immersion in the virtual world, so they prolonged the interaction.

 ISBN 978-80-86943-41-1

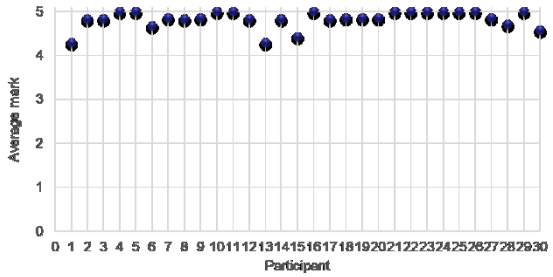


Figure 11 Graphical illustration of participants' average marks for research questions

6. CONCLUSIONS AND FUTURE WORK

As many researchers established, education is an area where VR technology can contribute to learning and teaching methods in a different way than traditional methods. These methods require interaction with a student by which he/she can develop creativity and innovativeness. In addition, an area which provides a lot of examples of abstract problems and where the VR potential can be utilized is mathematics. Because the use of VR is still not widespread, especially for educational purposes, the prototype of the elementary mathematical virtual classroom was developed. It is based on one of the ICT-AAC applications for learning basic mathematical concepts for younger children. That said, such an application does not require much skill or effort regarding spatial awareness as it is just a first step towards introducing VR for educational purposes to a wider mass. Prototype is then evaluated to analyze users' satisfaction with the application and the acceptance of the use of new technology in general. Learning in virtual environment in a natural and intuitive way provided by mathematical virtual classroom prototype goes beyond methods in formal education which require little or no interaction with students.

The results of user evaluation of mathematical virtual classroom showed that significant correlation between the duration of playing time and learning outcomes (in this case expressed as answers for a given mathematical problem task) does not exist but that the performance of a game depends exclusively on how an individual liked the immersion into and interaction in the virtual world. On the other hand, a certain correlation occurred between the results from objective and subjective measures for some participants. Participants who gave lower marks experienced unnatural behavior of virtual hands and had longer game play time than average. Other participants gave great marks even though they played the game for a long time which is an indicator that they enjoyed the virtual world and had no problems during the game. In general, the results of user evaluation showed overall satisfaction with the application which stems from the attractiveness of using modern technology. The results also showed that users are still not completely accustomed to the feeling of immersion into the virtual environment which stems from the fact that the concept of using virtual reality is still new. That is why further research is needed to make VR solutions as intuitive and understandable as possible in different domains.

For mathematical virtual classroom to be adopted as AAC system that children will be motivated to use, future research needs to be conducted. The initial user evaluation was conducted on participants without disabilities and the results were positive. Because of that, we have basis for further work where we plan to test the application with typically developing children as well as with children with disabilities to investigate their preferences and to see how they would accept the concept of learning in the virtual environment.

Encouraging feedback from the conducted study, which involved a mathematical application with a simple VR environment, reveals a potential for tackling new interesting research challenges in the follow-up study. For example, to be able to measure the possible impact of VR technologies in education, especially in mathematics domain, one can develop a more sophisticated VR application, intended for understanding and solving mathematical problems, that include spatial context. Also, future work will include an extensive user experience analysis of both versions of the application (i.e., the "*traditional*" *mobile version* and the *VR version*). Such an analysis will benchmark the two different ways of hand interaction mechanics: i) through a touchscreen (i.e. touch), and ii) through VR elements (e.g. grabbing and pointing). Ultimately, the analysis will shed a light on whether the VR systems are more engaging and more intuitive than the traditional methods of learning.

7. ACKNOWLEDGMENTS

This work has been supported by Croatian Science Foundation under the project 8813 (Human-centric Communications in Smart Networks of People, Machines and Organizations) and by Croatian Regulatory Authority for Network Industries under the project Looking to the Future 2020.

8. REFERENCES

[Ada07] Adamo-Villani, N. and Wright, K. (2007). SMILE: An immersive learning game for deaf and hearing children. In *ACM SIGGRAPH 2007 educators program (SIGGRAPH '07).* ACM, New York, NY, USA, Article 17

[Ala17] AlAwadhi, S., AlHabib, N. and Murad, D. Virtual reality application for interactive and informative learning. *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, Paris, 2017, pp. 1-4.

[Bab15] Babic, J., Slivar, I., Car, Z. and Podobnik, V. Prototype-driven software development process for

augmentative and alternative communication applications. *2015 13th International Conference on Telecommunications (ConTEL)*, Graz, 2015, pp. 1-8.

[Ded96] Dede, C., Salzman, M., and Loftin, B. (1996). MaxwellWorld: Learning complex scientific concepts via immersion in virtual reality. In *Proceedings of the 2nd International Conference on Learning Sciences*, pp. 22-29.

[Dun12] Duncan I., Miller, A., Jiang, S.: A taxonomy of virtual worlds usage in education. British Journal of Educational Technology, 43, 6 (2012), 949–964.

[Gri16] Grivokostopoulou, F., Perikos, I. and Hatzilygeroudis, I. An Innovative Educational Environment Based on Virtual Reality and Gamification for Learning Search Algorithms. *IEEE 8th International Conference on Technology for Education (T4E)*, Mumbai, India, 2016, str. 110-115

[Ict13] ICT-AAC project. (2013). ICT competence network for innovative services for persons with complex communication needs (ICT-AAC). Retrieved from: http://www.ict-aac.hr/index.php/en/

[Ict14] ICT-AAC Domino counter. (2014). Retrieved from: http://www.ict-aac.hr/index.php/hr/ict-aac-razvijene-aplikacije/android-aplikacije/domino-brojalica

[Kau09] Kaufmann, H. Virtual environments for mathematics and geometry education. *Themes in Science and Technology Education*, Special Issue: Virtual Reality in Education, vol. 2 (2009), 1-2; 131-152

[Lig07] Light, J., Page, R., Curran, J. and Pitkin, L. (2007). Children's ideas for the design of AAC assistive technologies for young children with complex communication needs. In *Augmentative and Alternative Communication*, 23, 4: 274-287.

[Lin16] Linares, R., Herrera, J. and Alfaro, L., "AliciaVR: Exploration of scientific articles in an immersive virtual environment with natural user interfaces," *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, Guayaquil, 2016, pp. 1-6.

[Mik11] Mikropoulos, T. A., Natsis, A.: Educational virtual environments: A ten-year review of empirical research (1999-2009). Computers & Education 56, 3 (2011), 769 –780.

[Pan10] Pantelidis, V. S. (2010). Reasons to use virtual reality in education and training courses and a model to determine when to use virtual reality. *Themes in Science and Technology Education*, 2(1-2), 59-70.

[Pti16] Pticek, M., Podobnik, V. and Jezic, G. (2016). "Beyond the Internet of Things: The Social Networking of Machines", *International Journal of Distributed Sensor Networks*, pp. 1-15

[Ray16] Ray, A.B. and Deb, S. Smartphone based virtual reality systems in classroom teaching – a study on the effects of learning outcome. In *2016 IEEE 8th International Conference on Technology for Education (T4E)*

[You98] Youngblut, C. (1998). Educational uses of virtual reality technology. *Technical report, IDA Document D-2128*. Alexandria, VA. Institute for Defense Analyses

 ISBN 978-80-86943-41-1

Two-phase MRI brain tumor segmentation using Random Forests and Level Set Methods

László Lefkovits
Sapientia University
Department of Electrical Engineering
Romania, Tg. Mureș
lefkolaci@ms.sapientia.ro

Szidónia Lefkovits
"Petru Maior" University
Department of Computer Science
Romania, Tg. Mureș
szidonia.lefkovits@science.upm.ro

ABSTRACT

Magnetic resonance images (MRI) in various modalities contain valuable information usable in medical diagnosis. Accurate delimitation of the brain tumor and its internal tissue structures is very important for the evaluation of disease progression, for studying the effects of a chosen treatment strategy and for surgical planning as well. At the same time early detection of brain tumors and the determination of their nature have long been desirable in preventive medicine. The goal of this study is to develop an intelligent software tool for quick detection and accurate segmentation of brain tumors from MR images.

In this paper we describe the developed two-staged image segmentation framework. The first stage is a voxel-wise classifier based on random forest (RF) algorithm. The second acquires the accurate boundaries by evolving active contours based on the level set method (LSM). The intelligent combination of two powerful segmentation algorithms ensures performances that cannot be achieved by either of these methods alone.

In our work we used the MRI database created for the BraTS '14-'16 challenges, considered a gold standard in brain tumor segmentation. The segmentation results are compared with the winning state of the art methods presented at the Brain Tumor Segmentation Grand Challenge and Workshop (BratsTS).

Keywords

Brain tumor, multimodal MRI, voxel-wise segmentation, random forest, level set method, feature selection, tumor structure, hierarchical segmentation, supervised learning.

1. INTRODUCTION

Early detection of diseases is of the utmost importance to maintaining or somehow regaining one's health, and thus it contributes to improving quality of life. The combination of various image processing techniques creates an efficient diagnostic tool. One part of the imaging techniques is built around automatic image segmentation, which is much faster than time-consuming analysis by experts.

Cerebral metastases usually become symptomatic in the form of headaches, focal neurological deficits or seizures, but they may also be found coincidentally in cancer staging scans. In any case, the earlier the tumor is detected, the better the chances of survival. In addition to sensitive automatic detection, precise segmentation of tumors is also required for efficient treatment and intervention planning. In particular, brain tumor segmentation consists of separating the different tumor tissues from normal brain tissue. Accurate and reproducible segmentation and characterization of abnormalities can be considered indispensable in medical diagnosis.

The subsequent sections of the paper are organized as follows: in section 2 the milestone approaches of the literature are summarized. In section 3 the first major stage of the proposed system, the random forest (RF), is described, followed by the mathematical details of the second stage, the level set method (LSM), in section 4. Finally, the results of our experiment (section 5) are presented with an emphasis on the improvement brought by the LSM. The performances obtained are compared to other systems and conclusions are drawn.

2. RELATED WORK

At present, there are many state-of-the-art brain tumor segmentation methods that have been developed. These have been implemented and published mainly for the Brain Tumor Image Segmentation Benchmark, organized yearly since 2012 [1]. There are two main categories: generative and discriminative models. Generative methods attempt to determine the probability distribution

function between the input and the target outputs. They rely on the Bayes theorem and are based on prior knowledge using appearance or anatomic properties. All these methods assume standardized data acquisition, registration and alignment in order to be converted into a generally usable probabilistic model [1]. On the other hand, discriminative models are capable of learning the classification function directly from a manually labeled training dataset. The main drawback is the requirement for a substantial amount of data in order to create sufficiently general and high-performing classifiers via supervised learning.

Today's leading architectures in the field of medical image processing and brain tumor segmentation are based on two major methods: the random forest decision tree ensemble [3] and deep learning via convolutional neural networks (CNN) [4].

Zikic et al. [5] combine a discriminative model using 40 decision trees in the classification ensemble with 2000 context-aware attributes, combining all of these with a generative model using tissue-specific probabilities for each patient.

Ellwaa et al. [6] create a random decision tree with an iterative approach using heuristics to gradually add the data from new patients to the training dataset.

Maier et al. [7] use the random forest classifier for the prediction of ischemic stroke lesion outcome. They include texture as anatomical features in the 200-tree ensemble.

Another radically different classification and segmentation approach is based on a state-of-the-art method called Deep Learning.

Chang proposed in [4] a very fast but highly accurate CNN architecture with few parameters. In this classification, the deepest convolutional output layers are combined with hyperlocal features from the input image.

Soltaninejad et al. [8] join the two methods. They utilized the VGG16 [9] fully convolutional neural net to obtain a feature map that is combined with a Gabor filter bank. All of these feature maps are fed to a random forest classifier.

The Level Set Method (LSM) proposed by Chan-Vese [10] is used to determine the active contour between two surfaces by minimizing the sum of intensity variance of the defined inner and outer regions. It is used for medical image segmentation only in combination with other segmentation methods [11, 12].

3. RANDOM FOREST

The random forest (RF) is an ensemble of decision trees suitable for the task of classification. It is one of the few methods applicable for a very large dataset, for example 3D medical images. Beside classification, it can also be used for feature selection because it estimates variable importance during the steps of the algorithm. The multitude of randomly generated decision trees representing the forest has very good generalization properties owing to the randomization process used in the construction of each tree. Each of the trees represents a unique weak classifier. The ensemble joins several such trees, thereby obtaining a strong classifier. The underlying database is randomly sampled with replacement and, for each tree, a different bootstrap set and out-of-bag (OOB) set is obtained. The bootstrap set is used in the creation of the tree. The OOB set (disjunctive to the bootstrap set) is used for evaluation purposes, for the computation of the generalized error of the ensemble. Not only are the data instances used randomized in each tree, but the splitting criterion of a tree-node is also based on randomness. Out of a large number (M) of variables (features) only a given number ($m_{tires}<<M$) are selected randomly for splitting. The optimum of the splitting criterion is computed only for these selected variables, based on the maximization of information gain. The OOB error is computed for each tree on the OOB set, using the tree structure obtained. The average OOB error of the ensemble is the unbiased estimator of the generalized error of the model (GE). [13]

The minimization of the generalized error involves the optimization of the RF parameters. The parameters which have to be tuned in order to obtain a well-working classifier are the number of trees in the ensemble (K_{trees}), the number of nodes in each tree (T_{nodes}) and the number of variables used as a splitting criterion in the nodes, called number of tries (m_{tries}).The number of trees (K_{trees}) influence the generalization error of the ensemble. If it is sufficiently large, the overfit of classification can be avoided, but the generalization error grows and the computation time increases. The number of nodes (T_{nodes}) is usually not limited in many of the other attempts in the literature. We have discovered that limitation is very important in order to avoid extremely deep trees. The third parameter is the number of variables (m_{tries}) randomly selected in each node. This value restricts the variables evaluated for finding the optimal split.

In our segmentation approach we make use of both the classification capacity of the RF ensemble and its variable importance measures applied in feature selection. The first step of creating the model is to fix a large number of low-level features (first order operators [mean, standard deviation, min, max, median, gradient]; higher order operators [difference of Gaussian, Laplacian, entropy, curvatures, kurtosis, skewness]; texture features [Gabor wavelets]; spatial context features [symmetry, projections,

 ISBN 978-80-86943-41-1

neighborhoods]), out of which the random forest is able to choose the most important ones. Only after this step does the training of the RF classifier described above follow, using the important features only. In statistical pattern recognition, the more adequate features are selected, the better the final decision will be. The RF approach offers an opportune method for the selection of relevant variables. In the case of RF, there are two possibilities to evaluate variable importance: Gini importance and permuted importance [13]. The variable importance depends on the RF ensemble obtained. Because the ensemble is based on randomness, the effective values of the importance are different for each new RF, but the order of important variables is, on average, similar. In our previous article we proposed a feature selection approach using the variable importance given by RF. Due to this algorithm, we managed to considerably reduce the number of initial variables (V) to a much smaller amount ($V_{imp}<V$), which are considered important with regard to brain tumor segmentation. The algorithm proposed consists of the following steps:

1. Create an RF ensemble for variable importance evaluation;
2. Considering the order of importance, eliminate the least important $p\%$ of variables.
3. If variables are sufficiently reduced, continue with step 4, otherwise repeat from step 1.
4. Create the RF classifier considering the remaining variables.
5. Evaluate the classification performances obtained.
6. Accept or reconsider the number of iterations (steps 1-3) based on the classification accuracy.

In our experiments we considered different values of $p\%$ and a different number of iterations. At first, we were able to reduce a large number of unimportant variables, but in the last stages, only a few. This depends on the classification performances of the RF ensemble obtained.

4. LEVEL SET METHOD

The accurate segmentation of MR images is a difficult task due to unclear or blurred dividing surfaces between tissues. The level set method is used with predilection because it performs better than other segmentation algorithms such as the gradient, threshold or clustering methods. The performances are explained by the fact that in the level set method, the global proprieties of image intensities matter more than local ones. The variant of the level set method try to find an active contour which delimitates the image regions and evolves in time during the segmentation process. For this task we adopted the Chan-Vese algorithm [10], which tries to find the active contour by energy minimization. Namely, the sum of the intensity variance of segmented regions is minimized. Thus, the best location of the contour is in the force equilibrium state in the force field of the image. Furthermore, the implicit formulation of the active contour provides certain remarkable features, such as topological flexibility, good numerical stability and straightforward extension of the 2D formulation to the n-dimension.

The segmentation task can be enunciated by finding a curve (C) that separates the image (Ω) into disjointed regions ($\Omega_1, \Omega_2, \ldots, \Omega_n$). Mathematically, this can be formulated to find the curve (C) which minimizes the Mumford-Shah functional:

$$\begin{aligned} &F(c_1,c_2,C) = \mu L(C) + \nu A(in(C)) + \\ &+\lambda_1 \int_{in(C)} |u_0(x,y) - c_1|^2 \, dxdy + \\ &+\lambda_2 \int_{out(C)} |u_0(x,y) - c_2|^2 \, dxdy \end{aligned} \quad (1)$$

where c_1 and c_2 are the average intensity levels inside and outside of the contour, $L(C)$ is the length of curve, $A(in(C))$ the area inside the curve, $u_0(x, y)$ image intensities and the μ, ν, λ_1, λ_2, parameters should be determined for each segmentation type.

In the level set formulation, instead of searching for the solution in terms of C, we are looking for a surface $\Phi(x,y)$ with the following properties:

$$\begin{aligned} &C = \{(x,y) \in \Omega : \Phi(x,y) = 0\} \\ &inside(C) = \{(x,y) \in \Omega : \Phi(x,y) > 0\} \\ &outside(C) = \{(x,y) \in \Omega : \Phi(x,y) < 0\} \end{aligned} \quad (2)$$

where $\Phi(x,y)$ is the signed distance function from C, 0 on curve C, negative outside Φ and positive inside Φ. The distance function $\Phi(x,y)$ evolves in time in such way that the curve C is the zero-level set of $\Phi(x,y,t)$

$$\begin{aligned} &F(c_1,c_2,C) = \\ &= \mu \int_{\Phi} \delta_0(\Phi(x,y)) |\nabla \Phi(x,y)| \, dxdy + \\ &+\nu \int_{\Phi} H(\Phi(x,y)) \, dxdy + \\ &+\lambda_1 \int_{\Omega} |u_0(x,y) - c_1|^2 H(\Phi(x,y)) \, dxdy + \\ &+\lambda_2 \int_{\Omega} |u_0(x,y) - c_2|^2 H(1 - \Phi(x,y)) \, dxdy \end{aligned} \quad (3)$$

 ISBN 978-80-86943-41-1

where δ is the Dirac function and *H* is the Heaviside function determining the inside (outside) of curve *C*. The first term is the length of the curve, the second is the area inside the curve, the third and fourth terms are energy terms inside and respectively outside the curve. Using the level set formulation, the image segmentation becomes an energy minimization problem, which leads to the solution with the corresponding Euler-Lagrange equation:

$$\frac{\partial \Phi}{\partial t} = \frac{\partial F}{\partial \Phi} \tag{4}$$

By using the Gateaux derivate of the energy function $\partial F/\partial \Phi$ we obtain the corresponding Euler-Lagrange equation:

$$\frac{\partial \Phi}{\partial t} = \delta(\Phi)\Big[\mu\kappa(\Phi)\big|\nabla\Phi\big| - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\Big] \tag{5}$$

where $\kappa(\Phi)$ is the curvature of Φ, $u_0(x, y)$ image intensities and the μ, ν, $\lambda 1$, $\lambda 2$, parameters should be determined for each segmentation type.. This partial derivate equation (PDE) can be easily solved with the standard gradient descent using variational methods. In this framework, the c_1 and c_2 are constant in the inside and outside region, respectively, and can be determined by

$$c_1(\varphi) = \frac{\int_\Omega u_0(x,y)H(\varphi(x,y))dxdy}{\int_\Omega H(\varphi(x,y))dxdy}$$
$$c_2(\varphi) = \frac{\int_\Omega u_0(x,y)(1-H(\varphi(x,y)))dxdy}{\int_\Omega (1-H(\varphi(x,y)))dxdy} \tag{6}$$

The c_1 and c_2 are the mean values of intensities in the segmented regions, inside and outside the curve *C*, respectively. It is desirable for these regions to be as homogeneous as possible. Taking this into account, we have to compute the level set function not on the whole image domain, but only in a narrow band near the different tumor tissue contours. This way, we managed to exploit the advantage of precise delimitation and at the same time reduce computation time.

5. RESULTS AND EXPERIMENTS

The primary task of segmentation is the delimitation of the tumor tissue from healthy brain tissue. At the same time, we also propose to determine the tumor structure by considering only four specific tissue types: the edema as well as three tumor substructures, which are the non-enhancing (solid) core, the enhancing tumor core and the necrotic (or fluid-filled) core [1]. These structures offer much more visual information for radiologists than a biological interpretation.

Our experimental setup utilizes the image database created for purposes of evaluating the approaches implemented participating in the BraTS Challenges (‘12-‘17) [2]. This database has become a gold standard in brain tumor segmentation during the last six years. The images were acquired in highly reputable clinic centers with different 1.5T or 3T MRI equipment, but strictly based on a standardized acquisition protocol. Experts in the field manually annotated the images using a segmentation protocol described in [14]. The manual annotation and segmentation of MR images is very time-consuming and requires fastidious and careful work even from an experimented specialist.

Each image set in the database consists of five types of registered images: T1, T1c (with the contrast material Gadolinium), T2, FLAIR and the expert-annotated image. Furthermore, the annotation contains four tumor classes: edema, enhanced tumor, non-enhanced tumor and necrotic core. The SICAS medical image repository [2] offers more than a hundred test image sets for evaluation, giving numerical performance results without showing the annotated image. In this online evaluation system there are only three classes which are taken into account and considered representative in clinical practice: Whole Tumor - WT (including all four tumor tissues), Tumor Core - TC (including all tumor structures except for edema) and Active Tumor - AT (only the enhancing core). The novelty of this article is the extension of our previous framework with a new stage in order to increase segmentation performances.

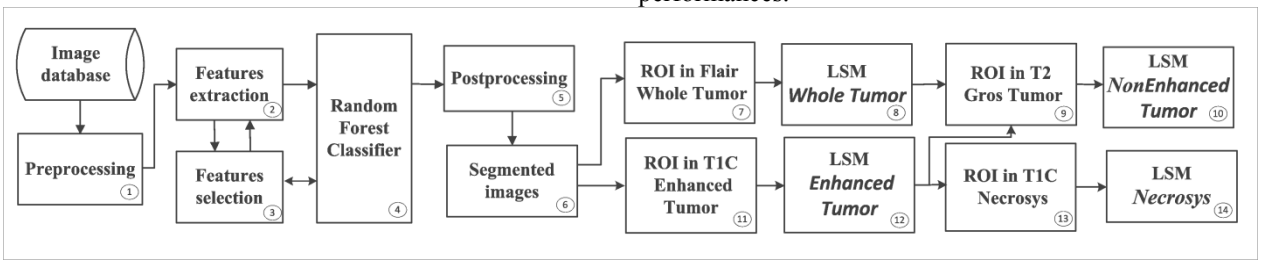


Figure 1. Block diagram of the proposed system

 ISBN 978-80-86943-41-1

The first stage of the framework proposed is a voxel-wise segmentation based on the random forest (RF) algorithm and is described in detail in our previous work [15]. The first stage corresponds to the blocks (1)-(6) in Figure 1.

The delimitation surface between tissues approximates the gold standard only roughly, and the internal tumor structure detected differs slightly from the annotation. In order to improve the segmentation results obtained after the random forest approach, our idea is to refine the contour of tumor tissues by applying the level set method. This method has two major drawbacks: it requires adequate initialization and is only capable of delimit nearly homogenous regions. The first drawback is overcome by considering the initial curve provided by the previous segmentation stage obtained from the RF approach. Secondly, we propose to determine the internal structure of the tumor in multiple steps starting from the inside towards the outside of the tumor. This layered detection of the different tumor tissues corresponds to the expect annotation protocol described in [14].

The primary assumptions of accurate medical image processing are the images without artifacts or noise. In addition, well-defined and repeatable correspondence between tissues and pixel intensities is also expected. In order to fulfill the desired criteria we applied three important correction procedures, in the following order: bias-field correction, noise filtering and intensity standardization in preprocessing.

For voxel-wise segmentation we transformed the image database previously described into a numeric database where each instance corresponds to a voxel, and the attributes are the values of several local image features. The problem is to determine the most significant features for the segmentation task proposed. In this field there is no recipe; every author defines the feature set based on their own experience or intuition. We defined *240* low-level image features in each image modality (T1, T1C, T2, Flair) and obtained a *960*-feature set ($V=4\times240$) that characterizes a voxel and its surroundings. However, a single 3D image from the database used contains about 1.5 million pixels; in our setup, the training database contains 50 brain images occupying about 500 GB of memory. Such a large database is practically unmanageable, and therefore we need to reduce it.

There are two ways of reducing this size: reducing the number of instances and/or the number of features. The number of instances can be reduced by random subsampling of the database. The number of instances belonging to the healthy brain tissue-class is ten times larger than the instances belonging to the tumor-class, and thus a sampling of 10:1 does not cause loss of information.

After this sampling of instances the database still remains large, and therefore it is necessary to reduce the number of features as well. Using the algorithm we proposed for variable importance evaluation, we managed to select the *120* most important features (V_{imp}) to be applied in this segmentation process. We showed that the OOB error obtained by the classifier build on this reduced feature set remains almost the same with the reduced set. The algorithm proposed in [15] uses the random forest variable importance evaluation and is able to run on the very large database.

The parameter optimization of the random forest and the methods applied for building a well-performing classifier for MR brain tumor segmentation is explained in our article [16]. Our optimized classifier is composed of $K_{trees} = 100$ trees, each having a size of $T_{nodes} = 2048$ nodes. The splitting criterion is evaluated with $m_{tries} = 9$ randomly chosen features out of the whole $M=120$ features/voxel. The classification results obtained on the BraTS 2016 test set are given in (Table 1, column 3).

The results obtained are comparable with the latest reported results (Table 1, columns 1-2), described in [1].

	BraTS 2012 [1]	BraTS 2013 [1]	Our RF classif.	Our 2staged classif.
WT	0.63-0.78	0.71-0.87	**0.75-0.86**	**0.80-0.91**
TC	0.24-0.37	0.66-0.78	**0.72-0.82**	**0.75-0.85**
AT	-	-	**0.78-0.84**	**0.82-0.88**

Table 1. Segmentation results

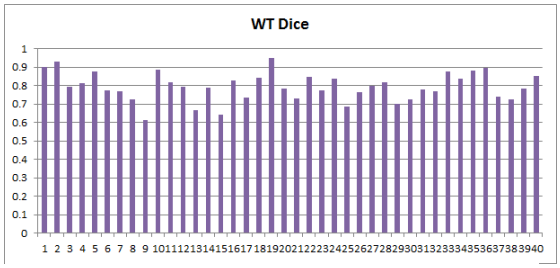


Figure 2. Dice coefficients of WT with RF

The results are shown (in Figure 2 and 3) for a randomly chosen 40 images from the test set having a mean of 0.793 Dice score on the whole tumor (WT) and 0.78 for the active tumor (AT) with a higher standard deviation (Figure 7 first and third boxes) .

ISBN 978-80-86943-41-1

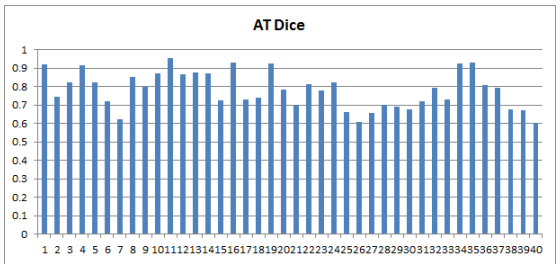


Figure 3. Dice coefficients of AT with RF

The results are also depicted graphically on a brain slice of two different images from the test set, (Figure 4). The green are the contour of the given annotation, the red are the RF segmentation results, the blue are the LSM segmentation results and the white are the ROI for LSM. We can see from these images that the delimitation surfaces between tissues are not sufficiently accurate and represent segmentation errors. It is obvious that a well-chosen local segmentation method should improve the results on the delimitation contours. Our idea was to exploit the advantages of the level set method in delimitating the borderlines of two regions belonging to two different tissues more precisely. In practice, this method may be predominantly used in the case of image zones with two tissues (Ω_1, Ω_2) and an initial approximate delimitation surface (representing a contour in plane - C) which must be used to initialize the regions in the level set method. The specification of such regions can be done by using a mask. The level set is applied only in the image domain (Ω) delimited by the given mask.

The segmentation protocol [14] states that "various tissue elements (edema, non-enhancing, enhancing, necrosis) usually follow an outside – inside sequence" and for one tumor-tissue "it is enough to always delineate what is outside". This structure is depicted in Figure 2 - a,b containing the expert annotation (black line) in T1c and T2 modalities.

Thus, as a second stage of segmentation, after the RF segmentation, we propose to apply the level set method according to these steps:

1. The edema region looks like a homogenous and hyperintense signal in Flair images and/or low signal in T1c (Figure 4a). To improve the delimitation surface of the edema from healthy tissue, we applied the level set in a ROI (region of interest) of the Flair images. This ROI is obtained by enlarging the edema region determined in RF stage by two morphological transformations. First we created conexzone of size 3 pixels and a ball type dilatation with radius of also 3 pixels. In this way we obtained a surface Ω_0 that includes all tumor structures in 99%. The Ω_0 is the ROI (block 7, Figure 1) where we search for the delimitation surface between the brain tissue and edema. The LSM segmentation we applied in this ROI (block 8 Figure 1) on Flair images in order to delimitate the whole tumor (WT) from the healthy tissues, being surface Ω_1 (Figure 4a).

2. We consider only the enhanced tumor, delimitated in the RF. Inside this ROI (block 13 Figure 1, $\Omega = \Omega_3 \cup \Omega_4$) there are only two tissues: the enhanced tumor (Ω_3), which is a brightly colored tissue in the T1c modality and the necrotic core (Ω_4) which is dark. The level set method is able to precisely delimitate the necrotic core (Ω_4), in T1c modality (Figure 4d).

3. The surface of the whole tumor Ω_1 obtained in the step 1, ($\Omega_1 = \Omega_2 \cup \Omega_3 \cup \Omega_4$) encapsulates all four tissues: edema with contour Ω_1 , non-enhanced tumor (contour Ω_2), enhanced tumor (contour Ω_3) and necrotic core (contour Ω_4). The previously segmented necrotic core (Ω_4) has already been segmented (step 2) and can be eliminated from ROI. Therefore, we apply the level set only in the remaining ROI (block 11 Figure 1, $\Omega = \Omega_2 \cup \Omega_3$) in order to find the delimitation surface of the enhanced tumor (Ω_3), which is brighter than the edema and non-enhanced in the T1c modality, (Figure 4b). The LSM stage delimitates the enhanced tumor surface Ω_3 more accurately then the RF stage (block 12 Figure 1).

4. With the surface obtained from the RF segmentation stage, the whole tumor ($\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \Omega_4$) encapsulates four tissues: edema (Ω_1), non-enhanced tumor (Ω_2), enhanced tumor (Ω_3) and necrotic core (Ω_4). The previously segmented zones ($\Omega_3 \cup \Omega_4$, steps 2-3) are excluded from the ROI. . So the considered ROI (block 9,10 Figure 1) contains only two tissues edema (Ω_1) and non-enhanced tumor (Ω_2). In the domain $\Omega = \Omega_1 \cup \Omega_2$ we apply the LSM in order to find the delimitation surface of the non-enhanced tumor (Ω_2) which is slightly brighter than the edema in the T1c modality. The elimination of the enhanced tumor (Ω_3) before the LSM segmentation of this step ensures a more precise segmentation of the non-enhanced tumor (Ω_2) contour (Figure 4c)..

Applying the procedure described above, we were be able to improve our segmentation performance by 3-7%, compared to the first stage (Table 1 columns 3-4). The other benefit of the two-stage segmentation is the more correct delimitation of necrotic zones, to which the RF voxel-wise segmentation only offered a weak solution. Improvement brought by the second stage was measured also in terms of Dice coefficients (Table 1-column 4). Figures 5 and 6 show the numerical results referring to the same test set and measuring the Dice scores on WT and AT tumor types.

 ISBN 978-80-86943-41-1

Green contour: ground truth, red RF segmentation, white ROI for LSM, blue LSM improvement

Flair T1 T1c T2 LSM

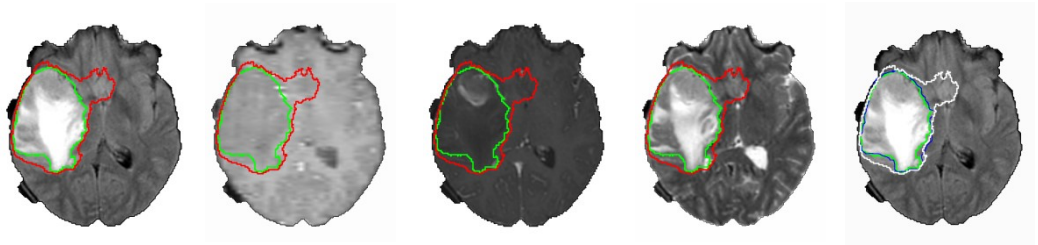


Figure 4a. Whole tumor (WT)

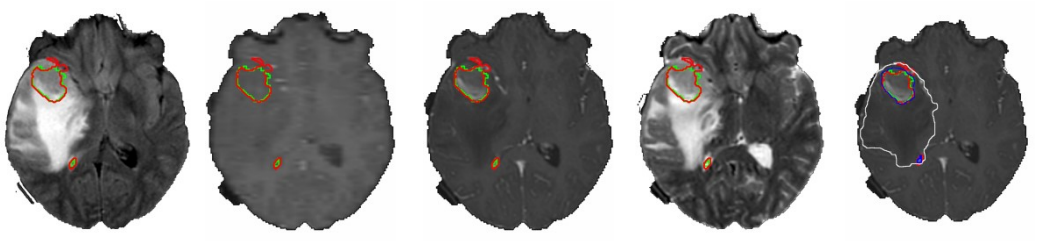


Figure 4b. Enhanced tumor (AT)

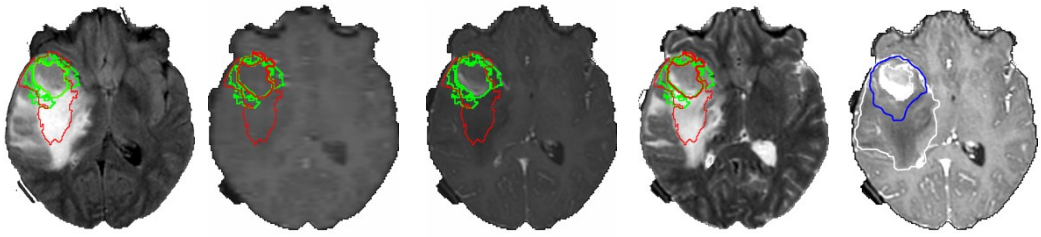


Figure 4c. Tumor core (TC)

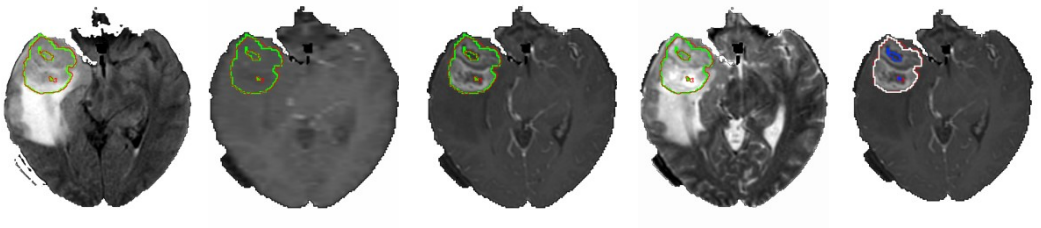


Figure 4d. Necrotic core (NC)

Figure 4. Visualized segmentation results on a brain slice

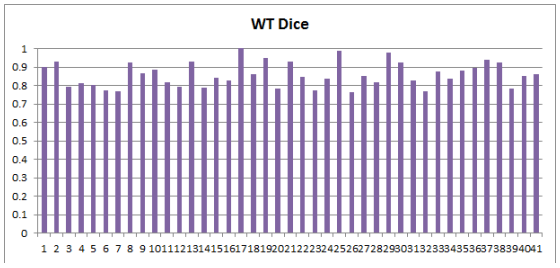


Figure 5. Dice coefficients of WT RF+LSM

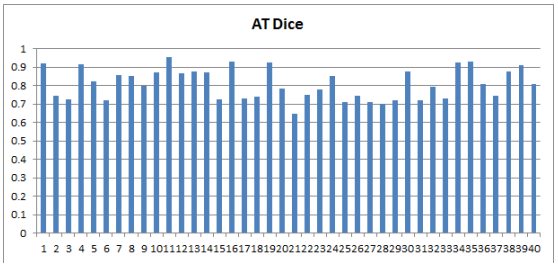


Figure 6. Dice coefficients of AT RF+LSM

ISBN 978-80-86943-41-1

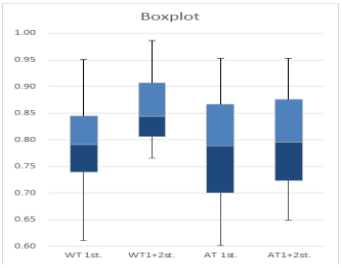


Figure 7. Boxplot comparison

The increased values are a mean of 0.854 for WT and a 0.806 for AT. These results are depicted in the boxplot also (2 and 4 boxes), to point out the standard deviation and the 1st and 3rd order quantiles (Figure 7.)

6. CONCLUSION

The novelty of this paper is the development of MR brain tumor segmentation framework obtained in two stages the random forest classifier linked with a well-defined sequentially applied contour refinement by the level set algorithm.

Firstly, the wise selection of features used and an adequate tuning of the random forest create a well-performing classifier for brain tumor segmentation. Secondly, the coarse segmentation obtained by the RF approach is merged with the level set with the aim of initializing its contours. Thus, we manage to further improve the precision of delimitation surfaces between neighboring tissues. Another important benefit of the proposed approach is the better determination of the tumor tissue structure, especially that of the necrotic core inside the enhanced tumor. For the future, we propose to implement a vector-wise LSM considering all modalities simultaneously applied in 3D MRI, instead of the current contour search run consecutively in 2D slices. Finally, it should be emphasized that accurate tissue delineation is difficult even for the well-trained eye of experts, and there are significant differences between experts' opinions. Although automatic segmentation is not always tantamount to perfection, it is much faster and reproducible, providing a useful tool in computer-aided medical diagnosis assistance.

7. ACKNOWLEDGMENTS

The work of L. Lefkovits in this article was supported by a grant of Sapientia Foundation – Institute for Scientific Research (KPI), P.N. 13/19/17.05.2017. The work of S. Lefkovits was supported by UEFISCDI grant no. PN-III-P2-2.1-BG-2016-0343, contract no. 114BG /01.10.2016.

8. REFERENCES

1. Menze BH, Jakab A, Bauer S, et al. The Multimodal Brain Tumor Image Segmentation Benchmark. IEEE Tr. Med. Imaging. 2015 34: p. 1993-2024.
2. "The SICAS Medical Image Repository". https://www.smir.ch/ BRATS/Start2015
3. Criminisi A, Shotton J. Decision forests for computer vision and medical image analysis: Springer Science & Business Media; 2013.
4. Chang PD. Fully Convolutional Deep Residual Neural Networks for Brain Tumor Segmentation. In MICCAI-BraTS; 2016. p. 108-118.
5. Zikic D, Glocker B, Konukoglu E et al. Context-sensitive classification forests for segmentation of brain tumor tissues. In MICCAI-BraTS 2012.
6. Ellwaa A, Hussein A, Al. Naggar E et al. Brain Tumor Segmantation Using Random Forest Trained on Iteratively Selected Patients. In MICCAI-BraTS; 2016. p. 129-137.
7. Maier O, Handels H. Predicting Stroke Lesion and Clinical Outcome with Random Forests. In MICCAI-BraTS; 2016. p. 219-230.
8. Soltaninejad M, Zhang L, Lambrou T, et al. Tumor Segmentation using Random Forests and Fully Convolutional Networks. In MICCAI-BraTS; 2017 Sep. p. 279-283.
9. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. 2014.
10. Chan TF, Vese LA. Active contours without edges. IEEE Transactions on image processing. 2001; 10: p. 266-277.
11 Zhao M, Lin HY, Yang CH, Hsu CY, Pan JS, Lin MJ. Automatic threshold level set model applied on MRI image segmentation of brain tissue. Applied Mathematics & Information Sciences. 2015; 9: p. 1971-1980.
12 Wu YT, Chen HY, Hung CI, et al. Segmentation of Hemodynamics from Dynamic-Susceptibility-Contrast Magnetic Resonance Brain Images Using Sequential Independent Component Analysis, WSCG, 2004; p. 267-274.
13. Breiman L. Random forests. Machine learning. 2001; 45: p. 5-32.
14. Jakab A. Segmenting Brain Tumors with the Slicer 3D Software Manual for providing expert segmentations for the BRATS.
15. Lefkovits L, Lefkovits S, Vaida MF. An Optimized Segmentation Framework Applied to Glioma Delimitation. Studies in Informatics and Control. 2017; 26: p. 203-212.
16. Lefkovits L, Lefkovits S, Szilágyi L. Brain Tumor Segmentation with Optimized Random Forest. In MICCAI-BraTS; 2016. p. 88-99.

 ISBN 978-80-86943-41-1

Advances for 3D printing: Remote control system and multi-material solutions

Adrián Luque Luque
University of Jaén, Spain
alluque@ujaen.es

Juan Manuel Jurado Rodríguez
University of Jaén, Spain
jjurado@ujaen.es

José Luis Cárdenas Donoso
University of Jaén, Spain
jcdonoso@ujaen.es

Francisco R. Feito Higueruela
University of Jaén, Spain
ffeito@ujaen.es

Abstract

Three-dimensional printing is an emerging manufacturing technology for many applications such as rapid prototyping, biomedical engineering and industrial designs. This paper aims to provide the implantation of a low-cost and extensible system in order to real-time monitor and modify 3D printing parameters. It is formed by a web-based platform and hardware components to manage and capture the printing process. In addition, we raise an overview about material properties in order to generate multi-material prototypes of bone tissue.

Keywords

3D printing, remote control system, multi-material, medical prototypes

1 INTRODUCTION

Additive manufacturing, commonly referred to as 3D printing has matured technically and is becoming more widespread for multiple application areas. Nowadays, the increasingly low cost and accessibility of three-dimensional (3D) printers has caused the exploitation of this technology from multiple enterprises. 3D printing is capable of creating physical models with a high quality in a short time. In this context, a lot of printing materials have raised which satisfy different physical properties depending on its subsequent application. In this paper, we are focused on Fused Deposition Modeling (FDM) technique [1]. FDM is the most widely used due to there are many low-cost 3D printers which support this technology and its printing parameters are easy to set. In this case, 3D printed objects are built by selectively deposition material in a pre-defined path layer-by-layer. However, FDM fabrication process has some limitations that must be known before any 3D printing. This technology belongs to the material extrusion family and can only be used with thermoplastic polymers through a filament form. Choosing the right type of material to print requires a high knowledge about its features and then, testing its behavior during the 3D printing. For this reason, we have used Octoprint application which provides a web interface for remote control of the 3D printer. It has been installed into a modular system that keeps a real-time feedback with information from different sensors fixed around the 3D printer.

Recently, the great increase of polymers, which are available in FDM technology, such as nylon, TPU (flexible) and soluble plastics makes possible the design of multi-material objects mixing different physical properties (e.g. mechanical, electrical, chemical, optical). Multi-material 3D printing provides challenges to allow 3D modeling of complex geometric structures and specific appearances. In this context, we are focused on the simulation of bone tissues.

In this paper, we first summarize the state of the art regarding 3D printing utilities and multi-material applications. Then, we describe the features of our control system for remote 3D printing and its hardware requirements. Afterwards we approach the study of multi-material objects for medical prototypes and finally, we discuss the pros and cons of the results.

The main contributions of this paper are:

- The implantation of a network platform to monitor and modify in real-time the 3D printing.
- A detailed study about multi-material features towards the 3D reconstruction of bone tissue.

2 RELATED WORK

According to the literature, there are different solutions for remote 3D printing [2]. However, the capabilities of these systems are limited during printing process. The total control for this task is a challenge which has not been resolved until now. The duration of 3D printing jobs depends on the model quality which is closely linked to its layer height. An optimal 3D printing must be supervised. For this reason, recently some web-based systems have been developed to satisfy printing controls. For instance, 3DPrinterOS or Waggle are remote controllers in order to review 3D printers behavior from multiple mobile devices. These under license solutions provide a remote 3D printing launching

and a web camera for real-time monitoring. A similar utility is offered by Repetier firmware [3]. It must be installed and configured for a specific 3D printer. Repetier is a standard open-source firmware which includes a web server in order to allow remotely some operation with the 3D printer through a web interface. However, this solution requires the installation of this specific firmware in the 3D printer.

In order to expand 3D printing and making it more versatile, we have also studied printing material properties for multi-material solutions [4]. In FDM technology, there are many extruder models that support multi-material printing such as E3D Kraken or Diamond HotEnd as well as 3D printers which have the capability for mixing different materials like Prusa MK3 or ZMorph among others[5].

In this paper we describe a remote control system for 3D printing which has been installed in a modular platform. It provides real-time monitoring and the capacity to modify some printer setting during printing process. In addition, we have studied the features of multi-material printing in order to simulate bone tissues.

3 REMOTE CONTROL SYSTEM

In our system we use a combination of open hardware and software, the core of the hardware is a mini computer, specifically a Raspberry Pi 3B, which is a very low cost system focused on teaching in developing areas. Due to its small size and versatility, it has become a key player in many technological projects, both industrial and domestic. Among its main advantages are the great connectivity through several ways(USB, Ethernet, GPIO, WiFi, Bluetooth, etc.), low power consumption, portability, and high performance computing.

In addition, some devices are used to extend the capabilities of the system. Some relays provide us the ability to switch on and off the printers, it is useful as a way to reduce electrical consumption and to launch an unexpected printing job. A web camera per 3D printer, allow us to see in real time the printing area, in order to check the status of the current model printing. The light sensor installed in the printing room, is capable of turning on a led lamp while the printers are running and there is no enough ambient light. For a better integration of our system, some models have been designed and printed (Figure 1).

In order to speed up remote printing, we have connected a relay to the Raspberry Pi that allows the printer to be switched on and off. This allows you to turn it off, to save energy when you finish a job, or to turn it on without having to go to the office. Several printer control with the same system: Running several Octoprint instances on the same device that listen on different HTTP ports. Each instance connects to the printer with its serial interface over USB. To expose those instances

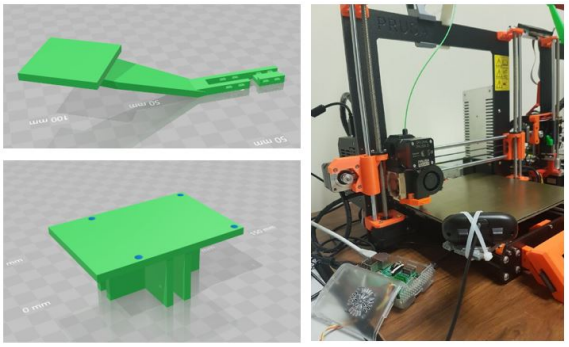


Figure 1: The implantation of our remote control system

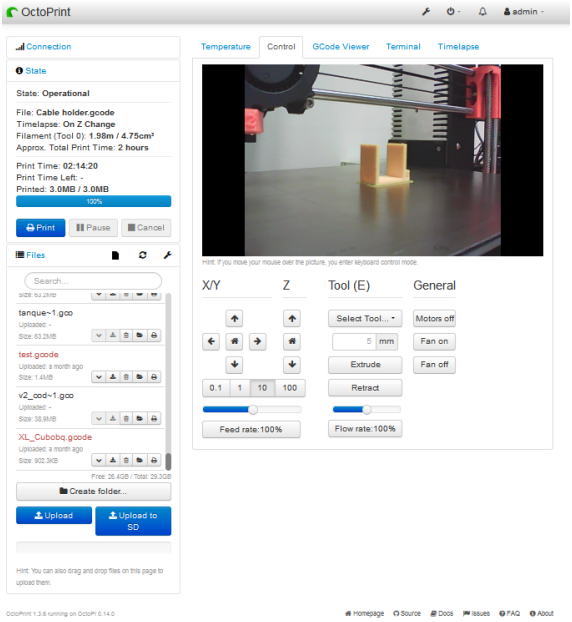


Figure 2: User GUI of Octoprint

to the world using standard HTTPS port, a proxy is used (NGINX). OctoPi is the basis of our system, it is a lightweight Linux distribution, derived from Raspbian. It contains the drivers and dependencies in order to work with almost all commercial 3D printers, and the OctoPrint (Figure 2). Octoprint is a 3D printing server which provides a web interface, allowing 24/7 network access. This software has the the following key features:

- Total control of the printer, axis movement, tools temperature, extruder behavior, etc.
- Real-time view of the printing area.
- Launch, pause, resume and cancel 3D printing.
- Serial communication with the 3D printer, using gcode commands.
- Time-lapse generation that allows to locate and solve printing errors.

ISBN 978-80-86943-41-1

4 ANALYSIS 3D PRINTING MATERIALS

Since last year, we have been testing a wide variety of 3D materials in order to know their physical properties and their behaviour after printing process. There are several main parameters which must be changed depending the material choice: printing temperature and velocity, infill density, retraction value, etc. [6]. The generation of multi-material models requires a deep analysis about material properties. In this paper, we are focused on the FDM printing materials with a diameter of 1.75 cm.

Many plastics have been tested and today, PLA (polylactic acid) is the best polymer for FDM printing. It has good mechanical properties, easy post-processing and allows faster 3D printings with a high quality. As a negative point, it begins to degrade at a low temperature, 60^{o}. PVA (polyvinyl acetate) is a water soluble substrate for 3D printing that allows for quick, inexpensive, and easy model separation. ABS (acrylonitrile butadiene styrene) is easy to machine and process after printing, with good mechanical properties, but difficult to print. It needs heated bed, emits harmful gases, and is prone to warping. HIPS (High Impact Polystyrene) has similar characteristics to ABS, it can be used as a support in 3D printing, as it dissolves in D-Limonene. But it is very difficult to print in conjunction with ABS, even at similar temperatures, the union layers between them do not adhere sufficiently. FILA-FLEX (mixture of polyurethane and other plastics) can creates parts with a high degree of elasticity and high tensile strength. On the other hand, it has many printing problems, and the printer's extruder may need to be modified. Hardened-NYLON: consists of a reinforced nylon, which gives it mechanical, thermal and abrasion resistance, ideal for industrial applications. But it is an abrasive plastic, it should be printed with steel nozzles and not brass.

5 MULTI-MATERIAL APPLICATION FOR MEDICAL PROTOTYPES

There are some FDM printers capable to print multi-material models by many techniques: (1) same materials with different color, (2) different materials, (3) a mix of them. A common practice is printing the support part of models with soluble materials. There are several ways to print multi-material models, the most basic one, is pausing the print in a predefined height and manually changing the plastic filament, but it is impracticable in our project, as it requires the constant attention of an operator.

A multi-extruder configuration, is a common solution to the multi-material printing, it allows to use different materials or colors in the same print. But in our testing, we discover two main disadvantages, multiple plastics can't be mixed at printing time, and the calibration phase is challenging, all extruders must be in the same exact height, a difference of a micron results in poorly printed models. The best setup tested is a 3D printer with a single extruder but multiple inputs, it has the benefits of the multi-extruder system, capable of mixing multiple materials, and the calibration is the same no matter how many inputs are present. With a 3 or 4 inputs printer, it is possible to mix different two colours in various proportions to create new colors, combine different plastic, or both. Some test with this configuration, conducted to a model printed with a mixture of ABS and HIPS (Figure 3), results in a printed part that has almost the same strength of the ABS, but it is more elastic.

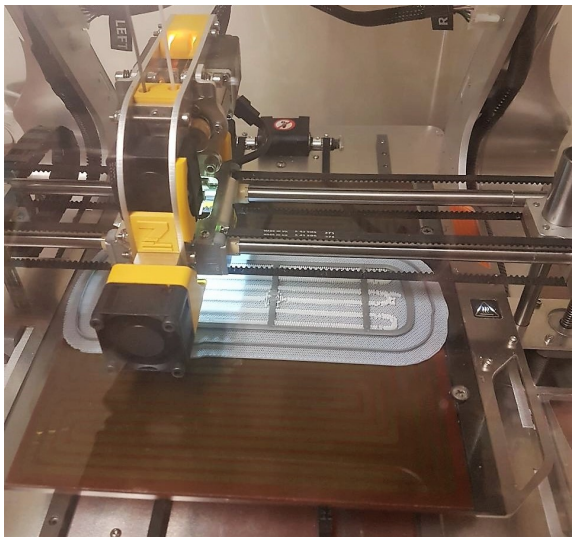


Figure 3: Multi-material prototype with Zmorph printer

This set of tests will serve to establish the basis for the creation of models that have the same mechanical characteristics as human bones. In addition, it provides support for a search project focused on the modeling of bone fractures. Multi-material prototypes play a key role in order to simulate physical properties, through different printing materials, of the trabecular and cortial regions of the bones.

6 CONCLUSIONS AND FUTURE WORK

3D printing industry has experimented a fast growing around multiple professional areas. Fused Deposition Modelling (FMD) technology allows the use of a wide variety of thermoplastic materials with the possibility of combining between them. In this paper, we have approached two 3D printing challenges: remote control system to monitor and modify in real-time the printing process and the study of material properties in order to generate multi-material solutions. 3D printing can also be used in medical applications such as bone tissue engineering.

 ISBN 978-80-86943-41-1

Over our current remote control system and the acquisition of a high knowledge about printing material properties, we are capable to model multi-material bones and explore their mechanical behaviors. This research opens many work lines for 3D printable complex structures with a real-time control during the whole process.

7 ACKNOWLEDGMENTS

This work has been partially supported by the Ministerio de Economía y Competitividad and the European Union (via ERDF funds) through the research projects TIN2017-84968-R and DPI2015-65123-R.

8 REFERENCES

[1] P. Dudek, "Fdm 3d printing technology in manufacturing composite elements," *Archives of Metallurgy and Materials*, vol. 58, no. 4, pp. 1415–1418, 2013.

[2] J. Martínez Arrieta, "Improvement and addition of features in 3d printing open source software octoprint," 2016.

[3] R. V. Aroca, C. E. Ventura, I. De Mello, and T. F. Pazelli, "Sequential additive manufacturing: automatic manipulation of 3d printed parts," *Rapid Prototyping Journal*, vol. 23, no. 4, pp. 653–659, 2017.

[4] P. Sitthi-Amorn, J. E. Ramos, Y. Wangy, J. Kwan, J. Lan, W. Wang, and W. Matusik, "Multifab: a machine vision assisted platform for multi-material 3d printing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 129, 2015.

[5] K. Vidimče, S.-P. Wang, J. Ragan-Kelley, and W. Matusik, "Openfab: a programmable pipeline for multi-material fabrication," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 136, 2013.

[6] X. Wang, M. Jiang, Z. Zhou, J. Gou, and D. Hui, "3d printing of polymer matrix composites: A review and prospective," *Composites Part B: Engineering*, vol. 110, pp. 442–458, 2017.

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

RDSpeed: Development Framework for Speed-Based Adaptation of Web Content on Public Displays

Amir E. Sarabadani Tafreshi, Adrian Wicki, and Gerhard Tröster
Wearable Computing Lab., ETH Zürich,
CH-8092 Zürich, Switzerland,
{tafreshi | troester} @ife.ee.ethz.ch

Abstract

Viewers of public displays perceive the content of a display at different walking speeds. While responsive design (RD) adapts web content to different viewing contexts, so far only the characteristics of the device and recently the proximity of the viewers are taken into account. Yet, little attention has been paid to speed-based adaption of content and its potential in case of public displays. We therefore decided to develop a framework that would support speed-based adaptation of public display applications. In this paper, we present a framework called RDSpeed that allows developers and designers alike to easily utilize our speed-based adaptation technique and integrate them into their own applications. RDSpeed extends the standard RD definition by adding new media queries for each adaptation technique. Media queries have long been established as the go-to technique for developing responsive web applications when dealing with a variety of different devices. A user study was conducted to investigate the potential of our content adaptation technique, and possible use and extensions in the future. We show several example adaptive applications of RDSpeed, as well as discussing advantages and limitations of our framework as revealed by our user study.

Keywords

Development framework, Responsive Design, Pervasive displays.

1 INTRODUCTION

The issue of engaging viewers of pervasive display systems (PDS) is a well-known problem [MWE+09]. The vast majority of PDS effectively disappear as people have become so accustomed to their low utility that they became highly skilled at ignoring them. Therefore, researchers are continuing to explore new forms of PDSs and studying their impact on users and user communities [STN17]. However, regardless of what content or services PDSs offer, their usability is limited on whether and how viewers perceive the content. Viewers' viewing experience depends on viewers' viewing context which can be utilized for adapting the content. Content adaptation not only can improve viewer's perception and viewing experience, but also enhance user engagement [TMN17]. The well-known issue that public display viewers ignore PDSs is promoting researchers to experiment with responsive design (RD) techniques to increase the utility and user engagement of PDS.

While responsive design adapts web content to various viewing contexts, viewer walking speed has not been taken into account as part of the viewing context in the case of public displays. It is important to note that viewers of public displays perceive the content at different walking speeds (see Fig. 1) which limit viewers' perception and content consumption. Humans have limited visual bandwidth and adapting content based on viewer walking speed can boost viewers' perception [Wic17].

In this paper, to enable researchers and developers to explore new forms of adaptation based on viewers' walking speed, we present a framework that supports the rapid development of web-based PDS applications featuring speed-based adaptation. The resulting Speed-Based-Responsive-Design (RDSpeed) framework extends media queries in the standard style sheet language used for describing the presentation of content (i.e., CSS[1]), to also consider the viewers' walking speed as part of viewing context in the case of public displays. To detect the number and speed of viewers, we use the camera-based Kinect sensing technology which is readily available as a commercial product. The framework is based on the standard CSS definition and builds on

[1] https://www.w3.org/standards/webdesign/htmlcss

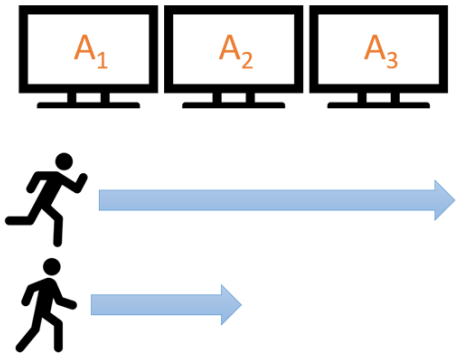


Figure 1: Demonstration of viewers passing in front of multiple displays at slow and fast walking speeds.

the Kinect SDK. Similar to responsive design breakpoints specified using CSS3[2] media queries, the framework supports a speed range concept which allows designers to specify more radical changes to the content and layout as viewers walk at different speed ranges.

The RDSpeed framework is characterised by four main aspects: (1) useful programming abstractions for the use of our proposed approach using extension of standard RD techniques (2) lightweight support for cross-platform, multi-device development based on native web technologies with which many developers are already familiar, (3) a flexible client-server architecture enabling adaptation of a variety of multi-display ecosystems, (4) an extensible architecture which allows multiple Kinect sensors to be used and connected to different distributed servers in order to cover wider viewing-field, and track more people in different locations and at different viewing angles.

We begin with a discussion of related work in Sect. 2, before presenting our speed-based adaptation model and the features of the RDSpeed framework in Sect. 3. The architecture and implementation are discussed in Sect. 4, followed by a description of the two adaptive sample applications developed using RDSpeed in Sec. 5. We then, in Sect. 6, report on an initial user study carried out to assess how our content adaptation techniques based on walking speed perform and how they are perceived when users experience them. We then present the results of the study in Sect. 7. The paper concludes with a discussion of our results in Sect. 8 followed by final remarks in Sect. 9.

2 BACKGROUND

Responsive web design (RWD) is an approach to web design to adapt sites to deliver an optimal viewing and interaction experience [Mar13]. With current RWD techniques, content can be adapted by considering device characteristics; but contextual factors beyond the device have so far received only little attention. By emergence of global networks of PDSs, the contextual factors are now playing an essential role in the adaptation [STN17]. Developers have used different techniques for RWD, and understanding these techniques would better clarify the appropriate techniques that could be extended to support additional contextual factors. Peterson [Pet14] gives a brief history of responsive web design where she states that in the year 2000 most websites were designed for a screen width of 800px. As screens grew wider over the years, developers would usually resort to fixed-width designs, so websites would look exactly the same on all screen widths. These fixed-width designs then led to the first mobile websites in the mid-2000s. Consequently, developers usually had to maintain two separate versions of a website — one for full-width desktop computer screens and one for much smaller mobile screens. As a consequence, most mobile websites were heavily stripped down versions of their full-width counterparts.

To deal with increasing variety of devices, web developers started to move to fluid instead of fixed-width layouts. *Flexible layouts* utilize flexible units such as percentages instead of absolute units such as points and pixels. Later, with the introduction of *media queries* and means of selecting and sizing images according to viewing context [Gar11], it was also possible to dynamically adjust a web layout to a given device screen. But it was not until 2010 that [Mar13] first introduced the concept of responsive web design which combined the concept of *media queries* with *flexible layouts*. *Media queries* are used to specify design breakpoints in terms of alternative CSS style rules to be applied in specific viewing contexts defined by values such as the device orientation, viewport size and pixel density [Fra15]. Both of these concepts are needed in order to create adaptive layouts which display content in a comfortable and visually appealing way on all available devices.

While an increasing number of practitioners advocate a mobile-to-desktop adaptation which is a mobile-first strategy in conjunction with responsive design, the focus in the research domain has mainly been on support for desktop-to-mobile adaptation [BN13]. However, only little attention has been paid in either practice or research to adaptation to very large displays, even though these are now common in (semi-)public spaces. However, one study investigated adaptation to large displays, particularly in the case of text-centric websites (e.g., online newspapers) [BMSN11]. In this work, the authors proposed a set of design metrics, developed an adaptive layout template and carried out a user study to show the benefits of the adaptation.

As the benefits of web technologies have been recognized in the PDS research community [TJS15], web architectures are mostly now the go-to sys-

[2] https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

ISBN 978-80-86943-41-1

tem design used for public display applications (e.g., [NEL16, NEL15, ESNL15]). Responsive design support is one advantage of web technologies for PDS applications and adapting the content can improve the user engagement [TMN17, STMT18, STBT18]. Dostal et al. [DKQ13, DHKQ14] also showed that adaptive interfaces are useful for addressing the user's various attention states. It's worth noting that content adaptation is passive, and unlike the interaction methods that require active interaction of users [RLC+06, LLCB08, BBKP07], happens automatically.

While it might be sufficient to consider the device size and the resolution for adapting the content to personal devices, it is important to also take other factors into account when adapting web content to public displays [TMN17]. Further, unlike personal devices that are easily accessible by users [STSTST17], some larger displays are out of reach. For instance, a study revealed that the average mobile smartphones viewing distance is about 12 inches [BRHH11], and therefore a user can easily adjust the distance to improve legibility if required. However, in the case of a public screen, users might require to significantly adjust their position or motion path, which they would only do if they were already engaged with the display. To address this gap, Tafreshi et al. [TMN17] proposed a model that integrates the proximity of viewers to a public display as an additional dimension of the viewing context considered in responsive design. This work also included a framework to support rapid prototyping for proximity-based adaptive display user interfaces. However, viewer proximity is just one factor when dealing with public displays; other factors such as viewer walking speed might be potential extension points for responsive PDS which gives researchers and developers the opportunity to study and apply new forms of adaptation in their PDS applications. As recent research [Wic17] showed, content adaptation based on walking speed improves the speed performance in searching tasks. Despite the advantages of walking speed-based content adaptation, little attention has been paid to use and study them for public display applications. One reason may be the lack of support offered to researchers and web developers.

3 THE RDSPEED FRAMEWORK

The main goal of the RDSpeed framework was to enable PDS designers and researchers to explore different styles and uses of speed-based adaptation on web-based PDS applications with focus on establishing design guidelines that could improve the public displays viewing experience in certain settings. RDSpeed provides an extension to the standard responsive design techniques (i.e., CSS) which provides a simple and familiar way for developers and designers alike to integrate our content adaptation technique into their own applications.

An initial starting point to build the framework was to identify the walking speed of viewers. To do so, it is needed to measure how quickly a viewer moves from one place to another. In principle, the speed (S) is a mathematical determination of the distance (D) a viewer moved divided by the time (ΔT). Likewise, we build our adaptation model given the $S = \frac{D}{\Delta T}$ formula. The distance that a viewer moved is based on the viewer position history – the previous (h) and current ($h+1$) position. Given the user position history in two dimensions of x-axis and y-axis, the viewers' walking distance (D) can be calculated using *Euclidean metric*. In two-dimensional Euclidean space, if $h = (x_h, y_h)$ and $h+1 = (x_{h+1}, y_{h+1})$ then the distance is given by

$$D = \sqrt{(x_h - x_{h+1})^2 + (y_h - y_{h+1})^2} \quad (1)$$

Accordingly, in our adaptation model, the resulting formula that calculates the walking speed as follows:

$$WalkingSpeed = \frac{\sqrt{(x_h - x_{h+1})^2 + (y_h - y_{h+1})^2}}{\Delta T} \quad (2)$$

Similar to how viewport size is often used in CSS3 media queries to define layout breakpoints, the calculated *WalkingSpeed* could be used to define design breakpoints in the form of media queries as well as for fluid web layouts. With respect to our adaptation model, these breakpoints correspond to a speed range so that content is adapted depending on the speed range in which a user is currently walking. As will be discussed later, in the case of multiple users walking at different speeds, there needs to be a strategy that determines which speed range to consider. We begin by describing the features and operation of the RDSpeed framework in terms of a single user.

The framework uses our model to compute the walking speed "walkingSpeed" at each time point. Accordingly, the framework provides the walking speed of the viewers which can be used for fluid design. In addition, a developer, in a customized setting, can partition the walking speed into multiple ranges defining the set of speed ranges.

Table 1 lists the key features that are encapsulated in the RDSpeed framework. *walkingSpeed* provides a number that shows the joint walking speed of current viewers of the display. *walkingSpeedRange* makes available a number containing the joint speed range of the viewers. In the case of multiple viewers, the speed and the range are computed based on the computation method

 ISBN 978-80-86943-41-1

Features	Examples	Description
Settings	RDSpeed ({bodyJoint: 'head', speedRange: [{'slow': 0, 'normal': 2, 'fast': 4}], multiUserAction: 'average'});	Set the body part to compute the walking speed, and the method to serve multiple viewers
walkingSpeed	body {font-size: calc(walkingSpeed * 10px);}	Register callback for the walking speed
walkingSpeedRange	@media (walkingSpeedRange: slow) {body{font-size: 12px;}}	Media query-based register callback for current walking speed range
numPeopleTracked	@media (numPeopleTracked >= 2) {#single { display: none; }}	Media query-based register callback for number of tracked viewers

Table 1: RDSpeed features with code examples. Callback media queries are based on the *settings* object.

Option	Description	Default Value	Possible Values
bodyJoint	which joint is used to position the body	'head'	'head', 'spine base', 'spine middle point', 'neck'
speedRange	speed breakpoints based on viewer's walking speed speed thr. N <= speedClassN < speed thr. N+1	['slow': 0, 'normal': 2, 'fast': 4]	array of names and thresholds [['SpeedName1', speed threshold1], ... , ['SpeedNameN', speed thresholdN]]
multiUserAction	computational method to serve multiple viewers	'average'	'average'; 'firstViewer'; 'majority';

Table 2: RDSpeed configurable setting options

configured in the framework. *numPeopleTracked* gives a number representing the number of current simultaneous viewers.

Settings is one of the key features of RDSpeed and allows the framework parameters and computation methods to be configured. The parameters together with their description along with the corresponding defaults and possible values are shown in Table 2.

bodyJoint defines what part of a viewer's body should be tracked to calculate the walking speed of a viewer. *speedRange* defines media queries for walking speed ranges. We adjusted the default value of this feature based on people's walking speed experiences[3] on treadmill i.e., slow (walking very slowly or standing still): 0-2mph, normal (walking at a normal everyday pace): 2-4mph, and fast (walking very quickly or running): above 4mph. *multiUserAction* offers some possible group handling methods to be able to serve groups of viewers an optimal view. The integration of this feature into the framework is because public displays are considered to often have multiple simultaneous viewers. The offered methods are devised from the feedback and discussions with the participants from the study which will be discussed in Sec. 6. These methods include: (1) the walking speed of the first detected viewer "*firstViewer*", (2) the average walking speed of individual viewers "*average*", and (3) the speed range with the most viewers "*majority*".

As shown in Fig. 2, to use the framework, the developer can change the constructed configurations in the RDSpeed setting object. They then have to write the style codes within the <*RDSpeedStyle*> tag, which allows to make use of special media-queries. The reason we did not use the normal <*style*> tag is because most modern browsers automatically attempt to correct invalid CSS code. In that case, browsers would remove the information not corresponding to the official CSS specification, such as the *walkingSpeed* keyword. Depending on the given parameters ("*bodyJoint, speedRange, MultiUserAction*") the content, layout or design of the display can be responsive. The corresponding media-queries and functions will be re-executed automatically by the RDSpeed framework every time new data arrives from the Kinect sensor. To simplify the definition of media queries based on the viewers' walking speed ranges, these breakpoints can be defined in the *speedRange* parameter of the *RDspeedSettings*. Then, as represented in Fig. 2, the framework provides the *speedRange* a viewer is moving at, based on the config-

[3] https://www.runnersworld.com/for-beginners-only/what-are-the-right-walking-and-running-speeds

 ISBN 978-80-86943-41-1

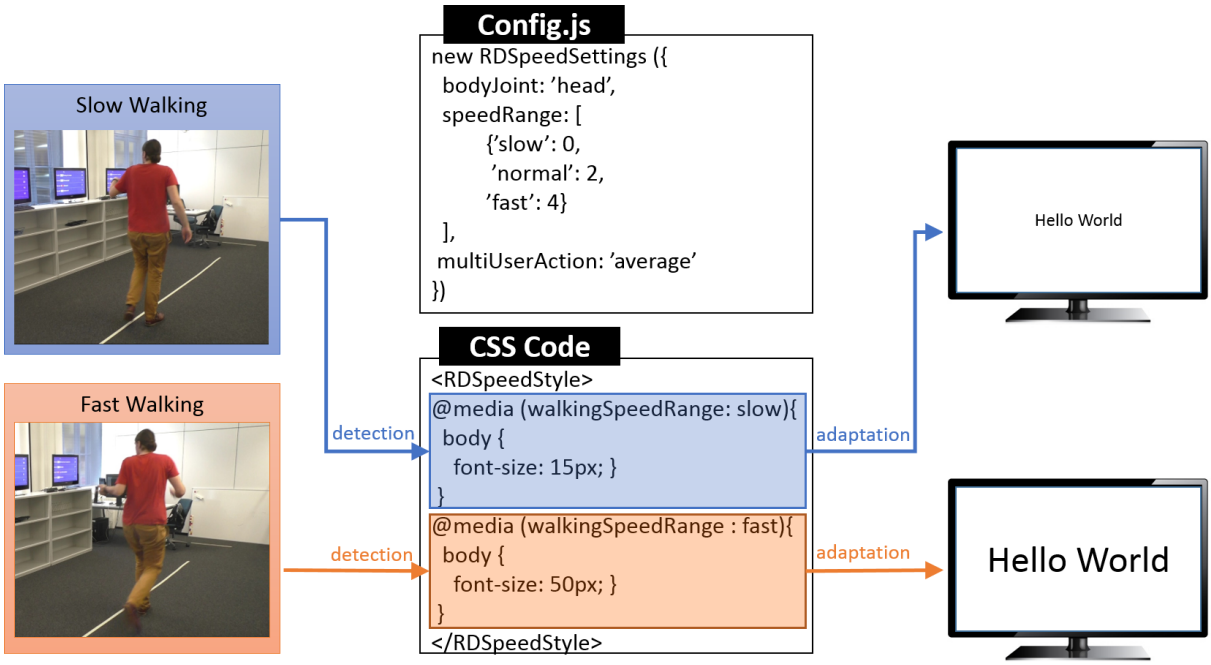


Figure 2: A sample use of the RDSpeed framework

uration in the *RDspeedSettings*. Likewise, the designer can choose what CSS style should be loaded for each walking speed range (*speedRange*). When considering a single viewer, the developer can extract the *walkingSpeedRange* information of that one viewer from the parameter *walkingSpeedRange*.

Using the computed parameters, the designer would also be able to make the adaptation fluid. To do so, they can use the *walkingSpeed* of a viewer. The *walkingSpeedRange* , *walkingSpeed*) parameters in the case of multiple simultaneous users, will be calculated using the preferred method that the developer can specify as part of the framework setting. Since the framework, in addition to the viewer walking speed and speed Range, provides more functionalities (i.e., the total number of the viewers), developers would be able to define different adaptations to handle single and multiple viewer(s).

4 ARCHITECTURE AND IMPLEMENTATION

Figure 3 illustrates the RDSpeed architecture. RDSpeed is based on a client-server architecture and consists of three main components – sensor, client and server-side components.

The server component forms the central backbone of the architecture and is written in Node.js. Node.js is a JavaScript runtime and uses an event-driven, non-blocking I/O model. The server component receives the pre-processed Kinect sensor data from the sensor components and aggregates them together. The server sends the aggregated data in real-time as a JSON-object to the connected clients using the Nodejs socket.io[4] library.

The client-side modules are responsible to parse the RDSpeed CSS code, extract the RDSpeed media queries and functions, receive the event notifications, and apply the developer's RDSpeed media queries and functions. The client-side module receive the new data from the server component, and then, using our model fed with the customized settings of the developer, they compute the arguments for the framework functions.

Sensor component is responsible for gathering the data from the *Kinect 2* device and relay the data to the server component. The data from the sensor component is acquired through the kinect2 Nodejs library[5] which provides access to the Kinect 2 data from the official Microsoft Kinect SDK[6]. The sensor component enables extension of the architecture with multiple Kinects which would allow more than six people to be tracked given the fact that a single Kinect can track a maximum of six people concurrently. It could also be used to handle speeds at different angles and/or in different locations. To do this, multiple sensor components can be connected to the server. One of other advantages of our architecture is that it enables scenarios in which the Kinect is not directly connected to the client computer. This includes the scenarios where there is cross-device interaction involving multiple dis-

[4] https://www.npmjs.com/package/socket.io
[5] https://www.npmjs.com/package/kinect2
[6] https://developer.microsoft.com/en-us/windows/kinect

ISBN 978-80-86943-41-1

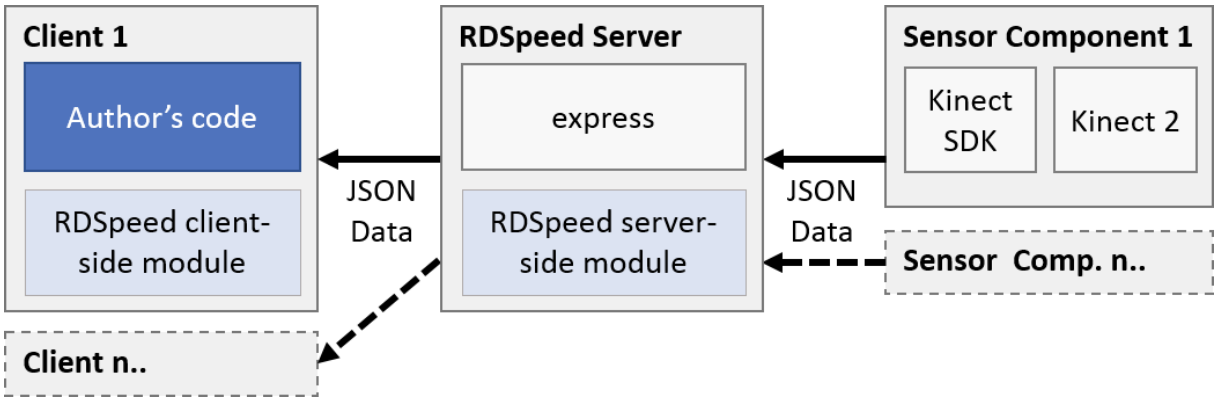


Figure 3: RDSpeed architecture.

tributed clients, which is possible with a single Kinect server [STST18]. Further, thanks to our cross-device web framework, our adaptation method can run across any kind of device.

5 APPLICATIONS

To define, demonstrate and test the capabilities of the RDSpeed framework, we now present two speed-based adaptive PDS applications created using our framework.

A first example application is a Train schedule application shown in Figure 4a. As the name suggests, the application shows departure times and destinations of trains. We chose this application because many people frequently use trains in their daily life and use public displays to look-up their train connections while some of them are in rush. We implemented two layouts with the intent of making one suitable for slow walking or still-standing viewers (see Figure 5a) and the other one suitable for fast walking viewers (see Fig. 5b). Here, our application design is so that a display only shows the necessary minimum information that is expected from a train schedule (i.e., train destination, platform, and departure time) in larger sizes to be better readable by fast-walking viewers. When a viewer stands or walks slowly, the display shows additional information such as occupancy and acceptability which requires reducing the font-sizes to show all information.

Our second example application, news application, was designed to increase awareness (see Fig. 4b). Our decision to create the adaptive news application was motivated by the fact that many people are curious about the current news while they are walking/running in front (semi-) public displays at different speeds. We believe that adaptation of current news content, particularly for emergency cases such as natural disasters or terrorist incidents, could help more people to be informed about the current situation. Here, in a context where a viewer is walking quickly or running, a display only shows the title of the news. In the case where the viewer walks normally or slowly, the display also shows the news abstract. If the viewer stands, the display shows the full news.

6 USER STUDY

We ran a lab study to get better insight on how our content adaptation technique based on walking speed perform and how they are perceived when users experience them. We opted for a lab study, which gives us control over the equipment and the environment, and the ability to directly interact with the participants and ask for clarification or feedback.

6.1 Participants

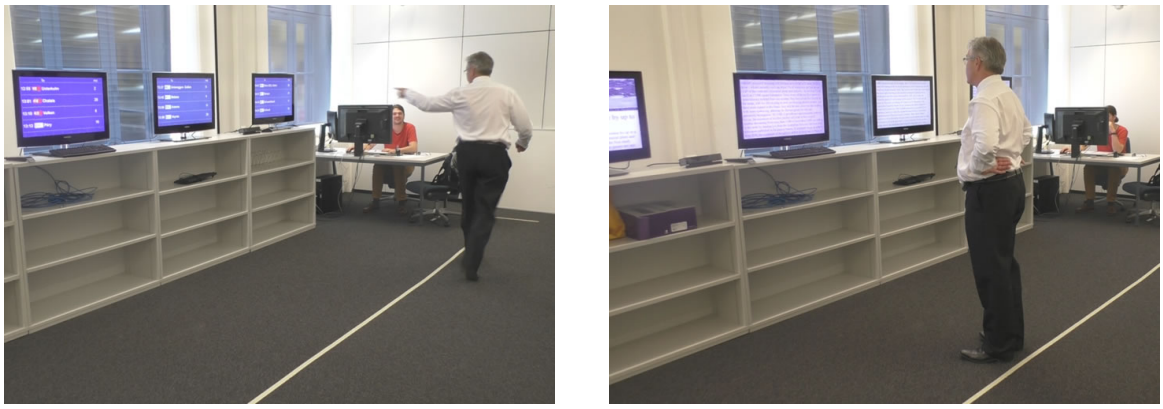
We recruited a total of 16 participants, 15 male, and one female. Most participants (14) had no computer science background. All of the participants either had normal or corrected to normal eye vision. Our participants were recruited using the *snow ball* sampling method at our University and also our social circle. 11 participants had age range of 25-34, two (18-24) , one 35-44, and two 55-64.

6.2 Setting

We used our lab's meeting room to install four 32-inch displays. The displays are placed on four shelves, each with a height of 1.2m. This ensures that the displays are located at eye-height for the majority of our participants. We introduced a small gap of approximately 60cm between the screens because this gives our participants more space when they have to walk in front of the setup at various speeds. We used two Kinect sensors to cover the space in front of the displays.

Figure 6 illustrates our study setup. To conduct the study, we opted for the train schedule application. This application is an example of an existing real-world application on public displays. In addition, most participants are already familiar with the layouts as they should have encountered similar applications at a train

ISBN 978-80-86943-41-1



(a) Train schedule (b) News application

Figure 4: The two adaptive sample applications: **(a)** Train schedule and **(b)** News application.

station or bus stop. The exemplary content generated by the application is implemented using the ScreenPress platform which is a platform developed for the rapid prototyping of PDS [STN17]. Each screen shows a total of 4 trains departing to randomly generated destinations. The destinations are chosen from a list of all Swiss municipalities[7].

6.3 Procedure and Methodology

We conducted the study with individual participants. Before they started the experiment, the participants were given a small introduction on what the system does and how they can expect it to adapt the content based on to their movement. We also asked for their consent to record the experiment using a video camera. We then enabled the walking speed-based content adaptation for the train schedule application and allowed the participants to freely experiment and explore the system. Afterwards, we asked participants to fill out a questionnaire and answer several semi-structured questions about their experience. The questionnaire first asked participants to provide demographic information about themselves before answering the questions about usability (easy and efficient to use), utility (functionality useful), stimulation (inspiration, wow experiences), value (importance), novelty, enjoyment, ease of learning, and responsiveness of the system. In addition, we asked several semi-structured and optional open ended questions about whether they encountered any technical issues or limitations of the system, the problems such a system might have when deployed to public settings, and the features they recommend to improve in the future system design as well as comments or suggestions.

Figure 5: The two different layouts of the train schedule application – **(a)** all additional information such as occupancy, train composition and icons are shown. To fit all information, the font-sizes have been decreased; **(b)** only the destinations, departure times and platform numbers are shown. As more space is available the font-sizes have been slightly enlarged compared to layout (a).

[7] Obtained from `https://www.bfs.admin.ch/bfs/de/home/grundlagen/agvch.html` accessed on the 06-05-2017.

7 RESULTS

The participants were able to freely explore the speed-based adaptive train schedule application.

Figure 7 shows how the participants rated the system according to novelty, utility, usability, value and importance, enjoyment and stimulation.

As can be seen in figure 7, the system was quite well accepted by our participants. Especially the novelty, utility (usefulness), usability, the value and importance are rated highly. However, the enjoyment and stimulation received a bit worse rating compare to other factors.

Participants gave the system an average overall rating of 6.375 (see Figure 8). The majority of participants (11/16) agreed or strongly agreed on the system ease of use. Three were neutral about the system's ease of learn. Most participants stated that, at first, the system was confusing, but after some time they were able to understand it and fully use its functionality.

Almost half of the participants (n=7) felt that the system was fairly or very responsive. Five participants stated that the system was somewhat responsive while four felt it was a little responsive.

ISBN 978-80-86943-41-1

Figure 6: Panorama view of the study setup.

7.1 Qualitative feedback

The feedback provided as comments gave us a better insight into the opinions of the participants.

P2 and P4 mentioned that they would prefer a system where only one screen is adapted (the one they stand in front of) instead of all the displays at once. P15 however, brought up a counter argument to the above statement: "*I liked, that all the screens changed, because I can see multiple screens from a single point of view. In a setting where only a single screen would change, I would have to actively move in order to see the detail content on another screen.*" P5 mentioned concerns on how the system would adapt to multiple viewers. P6 mentioned that instead of adapting while somebody is in front of the screens, it would be better to adapt before somebody enters the viewing field (to avoid confusion caused by the layout switch). P6 also stated that without prior instructions, the system might be confusing. P7 mentioned that he would like the system to support actual gestures (for zooming or moving content between screens). P12 showed interest "*to have an adaptation system on a cellphone. A system that recognizes walking speed and then adapts the displayed content.*" Such adaptation could be possible by making use of the accelerometer. P2 and P4 noticed a small uncovered angle between the two Kinect sensors where the adaptation system did not detect any movement. This issue points out one of the main limitations of the framework and the need for careful positioning of the sensors in the case of using multiple Kinects.

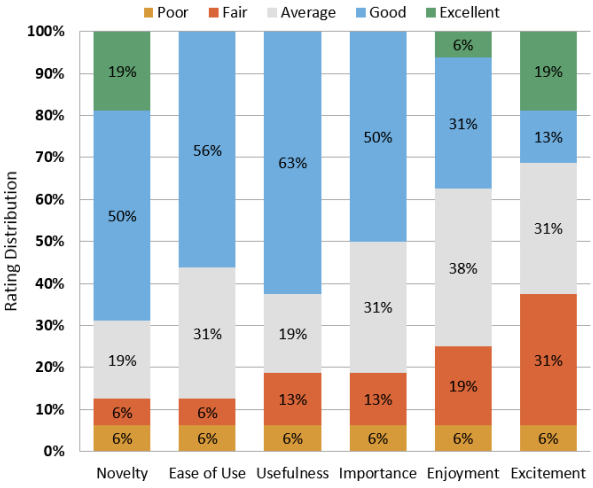


Figure 7: System adaptation rating of different factors.

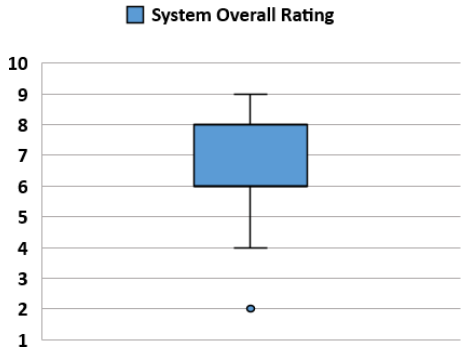


Figure 8: System overall rating

8 DISCUSSION

The participants highly rated the novelty, utility, usability, and value and importance of the system and rated the other factors such as enjoyment and stimulation a bit lower. A majority of the participants also stated that the system was easy to learn. In addition to the questionnaire feedback we were able to gather a plethora of qualitative feedback through conversations and observations. An important argument that was brought up by many participants was, that without prior instructions the system would be very confusing. Because it is not directly clear what the system does after users move in front of a display. A proposed solution to this specific problem was, to position the Kinect sensors before and after the displays, resulting in any adaptation happening before viewers are in front of the screens. This would additionally solve the problem of viewers getting confused or irritated by layout switches, as changes would occur before they are able to see the screens.

During the exploration phase many participants raised concerns regarding the suitability of the automatic

ISBN 978-80-86943-41-1

adaptation system for multiple users. Through discussions and conversations with the participants, we were able to devise three different solutions to dealing with multiple users. **First come, first adapted** The system only adapts to the first user who enters the viewing field of the sensors. The adaptation system then waits until no bodies are tracked anymore until it starts to adapt again (as soon as another user enters the viewing field of the sensors). **Majority adaptation** The system adapts to the majority of the users, i.e., adaptation is done according to whatever walking speed the majority of the current viewers exhibit. **Average adaptation** All calculated values are averaged over all tracked users and then the adaptation is done based on these values. We integrated these strategies into the RDSpeed framework as discussed in Sec. 3

9 CONCLUSION

We have presented a model which could be used to integrate the walking speed of viewers to a public display as an additional dimension of the viewing context considered in responsive design. In order to enable researchers and developers to explore new forms of adaptation based on viewers walking speed, we developed a framework that supports the rapid prototyping of speed-based adaptive applications. The framework was used to carry out a basic user study which allowed to get better insight on how our speed-based content adaptation technique perform and how users experience them. The feedback and the discussion with the study participants helped us to devise methods to handle multiple simultaneous viewers and extend our framework to support them. Now that we have the framework, we plan to experiment further with multi-viewer, multi-device settings in-the-wild, and investigate the potential benefits of speed-based adaptation of PDS.

10 REFERENCES

[BBKP07] Nadia Bianchi-Berthouze, Whan Kim, and Darshak Patel. Does Body Movement Engage You More in Digital Game Play? and Why? *Affective computing and intelligent interaction*, pages 102–113, 2007. DOI: 10.1007/978-3-540-74889-$2_1$0.

[BMSN11] M. Beneling, F. Matulic, L. Streit, and M. C. Norrie. Adaptive Layout Template for Effective Web Content Presentation in Large-Screen Contexts. *Proc. 11th ACM Symposium on Document Engineering (DocEng)*, 2011. DOI: 10.1145/2034691.2034737.

[BN13] M. Beneling and Moira C. Norrie. Responsive design and development: Methods, technologies and current issues. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7977 LNCS:510–513, 2013. DOI: 10.1007/978-3-642-39200-$9_4$7.

[BRHH11] Yuliya Bababekova, Mark Rosenfield, Jennifer E Hue, and Rae R Huang. Font Size and Viewing Distance of Handheld Smart Phones. *Optometry & Vision Science*, 88(7), 2011. DOI: 10.1097/OPX.0b013e3182198792.

[DHKQ14] Jakub Dostal, Uta Hinrichs, Per Ola Kristensson, and Aaron Quigley. SpiderEyes: Designing Attention-And Proximity-Aware Collaborative Interfaces for Wall-Sized Displays. In *Proc. 19th International Conference on Intelligent User Interfaces (IUI)*, 2014. DOI: 10.1145/2557500.2557541.

[DKQ13] Jakub Dostal, Per Ola Kristensson, and Aaron Quigley. Multi-View Proxemics: Distance and Position Sensitive Interaction. In *Proc. 2nd ACM International Symposium on Pervasive Displays (PerDis)*, 2013. DOI: 10.1145/2491568.2491570.

[ESNL15] Ivan Elhart, Federico Scacchi, Evangelos Niforatos, and Marc Langheinrich. ShadowTouch: A Multi-user Application Selection Interface for Interactive Public Displays. In *Proc. 4th International Symposium on Pervasive Displays (PerDis)*. ACM, 2015. DOI: 10.1145/2757710.2757735.

[Fra15] Ben Frain. *Responsive Web Design with HTML5 and CSS3*. Packt Publishing Ltd, 2015.

[Gar11] Brett S Gardner. The Spark of Innovation Begins with Collaboration. *Inside the Digital Ecosystem*, 11(1), 2011.

[LLCB08] Siân E Lindley, James Le Couteur, and Nadia L Berthouze. Stirring up Experience through Movement in Game Play: Effects on Engagement and Social Behaviour. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 511–514. ACM, 2008. DOI: 10.1145/1357054.1357136.

[Mar13] Ethan Marcotte. *Responsive Web Design*. Number 4 in . Editions Eyrolles, 2013.

[MWE+09] Jörg Müller, Dennis Wilmsmann, Juliane Exeler, Markus Buzeck, Albrecht Schmidt, Tim Jay, and Antonio Krüger.

Display Blindness: The Effect of Expectations on Attention towards Digital Signage. In *International Conference on Pervasive Computing*, pages 1–8. Springer, 2009. DOI: 10.1007/978-3-642-01516-8$_1$.

[NEL15] Evangelos Niforatos, Ivan Elhart, and Marc Langheinrich. Public Displays for Monitoring and Improving Community Wellbeing. In *Proc. ACM Intl Joint Conference on Pervasive and Ubiquitous Computing and Proc. ACM International Symposium on Wearable Computers (UbiComp/ISWC)*. ACM, 2015. DOI: 10.1145/2800835.2807954.

[NEL16] Evangelos Niforatos, Ivan Elhart, and Marc Langheinrich. Weatherusi: User-based weather crowdsourcing on public displays. In *International Conference on Web Engineering*, pages 567–570. Springer, 2016. DOI: 10.1007/978-3-319-38791-8$_5$0.

[Pet14] Clarissa Peterson. *Learning Responsive Web Design: A Beginner's Guide*. " O'Reilly Media, Inc.", 2014.

[RLC+06] Enrico Rukzio, Karin Leichtenstern, Vic Callaghan, Paul Holleis, Albrecht Schmidt, and Jeannette Chin. An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. In *International Conference on Ubiquitous Computing (UbiComp)*, pages 87–104. Springer, 2006.

[STBT18] Amir E. Sarabadani Tafreshi, Milan Bombsch, and Gerhard Tröster. Chained Displays: Configuration of Multiple Co-Located Public Display. *International Journal of Computer Networks & Communications (IJCNC)*, 10(3):27–44, 2018. DOI: 10.5121/ijcnc.2018.10303.

[STMT18] Amir E. Sarabadani Tafreshi, Kim Marbach, and Gerhard Tröster. Proximity-Based Adaptation of Content to Groups of Viewers of Public Displays. In *International Journal of Ubiquitous Computing (IJU)*, 2018. DOI: 10.5121/iju.2018.9101.

[STN17] Amir E. Sarabadani Tafreshi and Moira C Norrie. ScreenPress: A Powerful and Flexible Platform for Networked Pervasive Display Systems. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays*, page 13. ACM, 2017. DOI: 10.1145/3078810.3078813.

[STST18] Amir E. Sarabadani Tafreshi, Andrea Soro, and Gerhard Tröster. Automatic, Gestural, Voice, Positional, or Cross-Device Interaction? Comparing Interaction Methods to Indicate Topics of Interest to Public Displays. In *Frontiers in ICT*. Frontiers, 2018.

[STSTST17] Amir E. Sarabadani Tafreshi, Sara C. Sarabadani Tafreshi, and Amirehsan Sarabadani Tafreshi. TiltPass: Using Device Tilts As an Authentication Method. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS)*, pages 378–383. ACM, 2017. DOI: 10.1145/3132272.3134112.

[TJS15] Constantin Taivan, Rui José, and Bruno Silva. Web-Based Applications for Open Display Networks: Developers' Perspective. *International journal of computer systems science and engineering*, 30(1):21–30, 2015.

[TMN17] Amir E Sarabadani Tafreshi, Kim Marbach, and Moira C Norrie. Proximity-Based Adaptation of Web Content on Public Displays. In *International Conference on Web Engineering (ICWE)*, pages 282–301. Springer, 2017. DOI: 10.1007/978-3-319-60131-1$_1$6.

[Wic17] Adrian Wicki. Investigating Content Adaptation for Sequential Content Configuration of Chained Displays, 2017.

 ISBN 978-80-86943-41-1

Detection of change in video based on local pattern and photometric features

K L Chan

Department of Electronic Engineering, City University of Hong Kong
83 Tat Chee Avenue
Hong Kong, China

itklchan@cityu.edu.hk

ABSTRACT

The segmentation of moving objects in video can be formulated as a background subtraction problem – the detection of change in each image frame. The background scene is learned and modeled. A pixelwise process is employed to determine whether the current pixel is similar or not to the background model. The detection of change in video is challenging due to the non-stationary background such as illumination change, background motions, etc. We propose new features for background modeling. Perception-based local ternary patterns are generated from the same color channels as well as from different color channels. Features computed from the local patterns are stored in the background model as samples. If the current pixel is classified as background, the background model is updated. Finally, we propose a probabilistic refinement to improve each change region by taking into account the spatially consistency of image features. We compare our method with various background subtraction algorithms on some video datasets. Our method can achieve 13% better performance than other methods.

Keywords

Background modeling, Change detection, Local ternary pattern, Cross-channel pattern, Probabilistic foreground refinement

1. INTRODUCTION

Moving objects such as humans or vehicles, are often the focus of image sequence analysis. The segmentation of moving objects can be formulated as change detection. One common approach is to perform background subtraction. In that sense, the background scene is modeled. Change region is detected when it is found to be different from the background model. Sobral and Vacavant [Sob14a] presented a review and evaluation of 29 background subtraction methods. Background subtraction techniques can be categorized based on the features being used to model the background scene.

Statistical – Stauffer and Grimson [Sta00a] proposed modeling of background colors using mixture of Gaussian distributions (MoG). 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. Bouwmans [Bou11a] presented a survey on statistical background modeling.

Bag of visual words – Intensities or colors are sampled over a short image sequence. Elgammal *et al.* [Elg02a] proposed an algorithm for estimating the pdf directly from previous pixels using kernel estimator. Kim *et al.* [Kim05a] proposed to represent the background by codebooks which contain quantized background colors. Barnich and Van Droogenbroeck [Bar09a] proposed a sample-based background subtraction algorithm called ViBe. Background model is initialized by randomly sampling of pixels on the first image frame. Hofmann *et al.* [Hof12a] proposed a similar non-parametric sample-based background subtraction method with 9 tunable parameters.

Pattern – Recent research showed that modeling background by local patterns can achieve higher accuracy. 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 method using local binary similarity pattern (LBSP) estimated in the spatio-temporal domain. Ma and Sang [Ma12a] proposed the multi-channel SILTP (MC-SILTP), which is an improvement of SILTP, with pattern computed from RGB color channels. In [Cha16a], we proposed to model the background by perception-based local binary pattern.

2. SAME-CHANNEL AND CROSS-CHANNEL LOCAL TERNARY PATTERNS

We propose novel perception-based local ternary patterns which can be used effectively to characterize various dynamic circumstances in the scene. At each image pixel, patterns can be generated from the same color channels and different color channels. Figure 1 shows a block of 3 x 3 pixels. Each pixel of the block, n1 to n8, (except the center pixel) is compared with the confidence interval (*CI*) of the center pixel *b*. *CI*(*b*) is defined by (CI_l, CI_u) where CI_l and CI_u are the lower bound and upper bound of *CI* respectively. The pattern value *p* is set according to the following equation

$$p_k = \begin{cases} 00, & CI_l \le n_k \le CI_u \\ 01, & n_k > CI_u \\ 10, & n_k < CI_l \end{cases}, 1 \le k \le 8 \qquad (1)$$

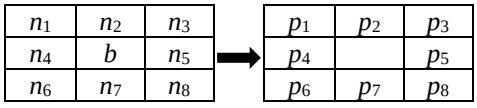


Figure 1. A block of pixels with center pixel *b* and neighbors n_1 to n_8 is transformed into ternary pattern.

The confidence interval *CI*(*b*) can be defined as ($b - d_1$, $b + d_2$). According to Weber's law [Gon10a], d_1 and d_2 depend on the perceptual characteristics of *b*. That is, they should be small for darker color and large for brighter color. Haque and Murshed [Haq13a] derived the linear relationship $d_1 = d_2 = c * b$, where *c* is a constant. We adopt the human visual perception characteristics in transforming pixel colors into local ternary pattern. *CI*(*b*) is defined as ($b - c_1 b$, $b + c_2 b$). Using peak signal-to-noise ratio (PSNR) measure, *b* and $b - c_1 b$ are just perceptually different from each other if

$$20\log_{10}\frac{I_{\max}}{b - c_1 b} - 20\log_{10}\frac{I_{\max}}{b} = T_p \qquad (2)$$

where $I_{\max}$ is the maximum intensity and T_p is the perceptual threshold. Similarly, *b* and $b + c_2 b$ are just perceptually different from each other if

$$20\log_{10}\frac{I_{\max}}{b} - 20\log_{10}\frac{I_{\max}}{b + c_2 b} = T_p \qquad (3)$$

To determine c_1 and c_2, the equations are simplified.

$$c_1 = \frac{10^{\frac{T_p}{20}} - 1}{10^{\frac{T_p}{20}}} \qquad (4)$$

$$c_2 = 10^{\frac{T_p}{20}} - 1 \qquad (5)$$

Assume T_p is 0.5 dB, $c_1 = 0.0559$ and $c_2 = 0.0593$.

If the center pixel *b* and neighbors n_1 to n_8 are from the same color channel, the Same-Channel Pattern (*SCP*) is generated as follows:

$$SCP_k = \oplus\{p_k^c\}, c = \{R, G, B\}, 1 \le k \le 8 \qquad (6)$$

where p_k^c is the binary pattern value for color channel *c* at position *k*, and $\oplus$ is the concatenation of the corresponding binary pattern values for all color channels. We choose the RGB color model instead of other color model such as YIQ because that avoids more computation in transforming the pixel values. If the center pixel *b* and neighbors n_1 to n_8 are from different color channels, the Cross-Channel Pattern (*CCP*) is generated as follows:

$$CCP_k = \oplus\{p_k^{c_b:c_n}\},$$
$$c_b{:}\,c_n = \{R{:}\,B, G{:}\,R, B{:}\,G\}, 1 \le k \le 8 \qquad (7)$$

where $p_k^{c_b:c_n}$ is the binary pattern value at position *k* estimated with center pixel *b* from color channel c_b and neighbor n_k from color channel c_n, and $\oplus$ is the concatenation of the corresponding cross-channel binary pattern values. One advantages of our local ternary patterns is that they are estimated from all color channels which can provide a more informative characterization of local image texture. Also, in flat image regions, the features derived from gray values or the same color channel may not be distinctive. This limitation can be alleviated with the use of *CCP*.

3. BACKGROUND MODELING AND CHANGE DETECTION

A number of image frames in each video are allocated for background model initialization. With this short image sequence, the local patterns are transformed into concise representation and stored as background samples. Also, the original color values are saved as photometric features in the background model.

First, *SCP* and *CCP* are combined into a 12-bit string:

$$CP_k = SCP_k \oplus CCP_k, 1 \le k \le 8 \qquad (8)$$

 ISBN 978-80-86943-41-1

For convenient of storage, CP_k is transformed into a single value:

$$F_k = \sum_{p=1}^{12} CP_k(p) \cdot 2^p, 1 \le k \le 8 \quad (9)$$

where $CP_k(p)$ is bit p of CP_k. Therefore, at each image pixel location, the local ternary patterns are transformed into an 8-dimensional feature vector $\{F_k\}$ and saved in the background model as one sample.

Change regions are detected by comparing each pixel of the image frame with the background model. It is a background/foreground segregation process. If features of the pixel match with the background model, it is a background pixel. Otherwise, it is a foreground (change) pixel. We adopted the sample-based approach. At each pixel of the current image frame, *SCP* and *CCP* are computed using the method as described in section 2. If the local ternary patterns and photometric features match with one sample of the background model, the current pixel is classified as background pixel. The similarity between patterns of the current pixel and the background model can be computed by measuring the Hamming distance between two bit strings

$$d_p = \sum_{k=1}^{8} \left| CP_k^i \otimes CP_k^B \right| \quad (10)$$

where CP_k^i is CP_k of the current pixel, CP_k^B is CP_k of a background sample, $\otimes$ is the XOR operator, $|\cdot|$ is the cardinality. The two sets of local ternary patterns are considered as similar if $d_p < \varepsilon_p$.

The similarity between the current pixel color and the color of the background sample is computed by

$$s_c = \frac{C^i \cdot C^B}{\|C^i\|_2 \cdot \|C^B\|_2} \quad (11)$$

where C^i is color the current pixel, C^B is color of a background sample, $\|\cdot\|_2$ denotes the Euclidean length of a vector. The two colors are considered as similar if $s_c > \varepsilon_c$.

The background model is updated alongside with the change detection. If the current pixel is classified as background, the features of the matched background sample will be replaced by the features of the current pixel.

4. FOREGROUND REFINEMENT

The background/foreground segregation result may contain false positive and false negative errors. For instance, isolated scene pixels may have features deviate from the background model due to illumination change or background motion. As they are not connected to form a region, they can be discarded without affecting the detection of real moving objects. Therefore, foreground regions less than 15 pixels are eliminated. The remaining foreground regions may have holes. The silhouette of the change region may be distorted. These false negative errors are usually caused by the similarity of the image features in the change region to the background model. We analyze the spatially consistency of image features and refine the change region probabilistically. Let *x* be a foreground (*FG*) pixel. Its neighboring background (*BG*) pixels *y* are defined by

$$y \mid dist(x, y) < D, x = FG, y = BG \quad (12)$$

where $dist(\cdot)$ is the city-block distance and *D* is fixed as 1. *y* are changed to *FG* when they have image features more similar to neighboring *FG* pixels than neighboring *BG* pixels. To analyze the local ternary pattern feature

$$y_i = FG \quad \text{if } log \frac{P(y_i=FG)}{P(y_i=BG)} > T_{f1} \quad (13)$$

$$P(y_i = FG) = exp\left(-\sum_j \left| d_p^{y_j} - d_p^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \quad (14)$$

$$P(y_i = BG) = exp\left(-\sum_j \left| \mu(d_p^{y_j}) - d_p^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \quad (15)$$

where $\mu(\cdot)$ is the mean of the local ternary pattern features in the background model. To analyze the photometric feature

$$y_i = FG \quad \text{if } log \frac{P(y_i=BG)}{P(y_i=FG)} > T_{f2} \quad (16)$$

$$P(y_i = FG) = exp\left(-\sum_j \left| s_c^{y_j} - s_c^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \quad (17)$$

$$P(y_i = BG) = exp\left(-\sum_j \left| \mu(s_c^{y_j}) - s_c^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \quad (18)$$

where $\mu(\cdot)$ is the mean of the photometric features in the background model. A false negative pixel will be

 ISBN 978-80-86943-41-1

corrected when both Equations (13) and (16) are satisfied.

5. RESULTS AND DISCUSSION

We evaluated the performance quantitatively in terms of F-Measure (F1). We compared our method with other background subtraction algorithms on two publicly available datasets. We selected sample-based method ViBe [Bar09a], pattern-based methods SILTP [Lia10a] and MC-SILTP [Ma12a] for comparison. Based on sample consensus, ViBe can achieve very good results with very few tunable parameters. ViBe uses RGB color model and a fixed spherical distance of 30 in matching new pixel with background samples. It keeps 20 background samples and the new pixel is identified as background with 2 matches. SILTP employs scale invariant local patterns. MC-SILTP is one latest pattern-based method and can perform better than SILTP. We implemented SILTP with the same set of parameters as reported in [Lia10a]. The only parameter value which was not mentioned is the number of training frames. Through experimentation, we find that the number of training frames is best fixed as 150. Similarly, we implemented MC-SILTP with the same setting as reported in [Ma12a]. As for our method, the first 50 image frames of the video are used for background model initialization. Other parameters are: $\varepsilon_p = 16$, $\varepsilon_c = 0.9$, $T_{f1} = 2.0$, $T_{f2} = 0.1$.

The Wallflower dataset [Toy99a] contains 6 videos. Each video comes with 1 manually labeled ground truth. The image frame size is 160 x 120 pixels. The dataset contains videos exhibiting gradual illumination change (TimeOfDay), sudden illumination change (LightSwitch), similar background and object color (Camouflage), moving background elements (Waving Trees), etc. Table 1 shows the F1 results of our method, ViBe, SILTP and MC-SILTP. The best result in a given row is highlighted. No method can achieve the highest F1 on all videos. Our method can achieve highest F1 on 5 videos. Overall, texture-based methods perform better than ViBe. Our method achieves the highest average F1 which is 13% higher than the second best method MC-SILTP. Also, our method can achieve consistently high F1 as indicated by the lowest variance.

Figure 2 shows the visual results. In "Bootstrap", humans already exist in the initialization image sequence. ViBe produces more false negative errors. Our method and SILTP relatively have lesser false negative errors. Our method also has lesser false positive errors than MC-SILTP. In "Camouflage", the difficulty is that the monitor and the clothing have similar color. Therefore, ViBe, SILTP and MC-SILTP produce many false negative errors. With probabilistic refinement, our method can drastically reduce false negative error. In "ForegroundAperture", the human remains stationary and stooped over the desk for some time. Features of the human are included in the background model. When the human rises, all methods produce false negative errors. In "LightSwitch", ViBe cannot adapt to the sudden change of light. Other methods can quickly respond. In "TimeOfDay", the room is very dark at the beginning. The light is turned on gradually and a human enters the room. SILTP and MC-SILTP cannot adapt to the change and result in large amount of false positive errors. ViBe performs better but the detected human is small. Benefit by the local ternary pattern features, our method can detect a larger human. In "WavingTrees", ViBe and SILTP produce many false positive errors in the trees behind the human. MC-SILTP still produce moderate amount of false positive error. Our method is quite effective in identifying the waving trees as background. In summary, our method can achieve a consistent and accurate performance under various kinds of complication in the background scene.

Sequence	**Our method**	**ViBe**	**SILTP**	**MC-SILTP**
Bootstrap	**0.846**	0.478	0.766	0.740
Camouflage	**0.966**	0.931	0.927	0.896
ForegroundAperture	0.768	0.644	**0.849**	0.665
LightSwitch	**0.759**	0.159	0.730	0.745
TimeOfDay	**0.678**	0.394	0.175	0.181
WavingTrees	**0.965**	0.933	0.712	0.946
Average	**0.830**	0.590	0.693	0.695
Variance	**0.014**	0.095	0.071	0.075

Table 1. F1 results on the Wallflower dataset

The Star dataset [Li04a] contains more challenging videos. Each video comes with 20 manually labeled frames as ground truths. The videos have different image frame size, from 160 x 120 pixels to 320 x 256 pixels. We selected ViBe [Bar09a], MC-SILTP [Ma12a], statistical method MoG [Sta00a], and pixelwise LBP (LBP-P) [Hei06a] for comparison. Table 2 shows the F1 results. The numeric results of MoG and LBP-P are from [Lia10a]. Our method can achieve highest F1 on 6 videos and second best on 2 videos. Overall, our method achieves the highest average F1 than all comparing methods which is 5% higher than the second best method LBP-P. Also, our method has the lowest variance.

Figure 3 shows the visual results of our method, ViBe and MC-SILTP. Some videos contain busy human flows (AirportHall, Bootstrap, Escalator, ShoppingMall). "Curtain" has a slowly moving curtain in the background. "Fountain" and

 ISBN 978-80-86943-41-1

"WaterSurface" contain moving water. In "Lobby", the light is dimmed and turned on later. "Trees" has waving trees and banner in the background. In "AirportHall", all methods produce false negative errors. ViBe and MC-SILTP have more false positive errors than our method. In "Bootstrap" and "Curtain", ViBe produces more false negative errors while MC-SILTP produces more false positive errors. Our method can detect a fairly good shape of the humans. "Escalator" is a difficult video. All methods many false positive and negative errors. In "Fountain", ViBe cannot detect the humans completely. MC-SILTP produces many false positive errors. Our method can effectively model the fountain as background and detect the humans. In "ShoppingMall", the main difficulty is the shadow. That causes more false positive errors in ViBe. Pattern-based methods can tackle shadow much better as can be seen in the shape of the humans detected by our method. In "Lobby", ViBe and MC-SILTP cannot adapt to the dimming of light and produce many false positive errors. Our local ternary pattern features have no problem in characterizing the illumination change. ViBe and MC-SILTP produce large amount of false positive errors in "Trees". ViBe also cannot detect the bus completely. Our method can effectively treats the waving trees as background and window of the bus as change region. In "WaterSurface", ViBe fails to detect the legs. Pattern-based method can model the water surface. However, due to similarity between clothing and water, they produce many false negative errors (holes) within the change region.

Sequence	Our method	ViBe	MC-SILTP	MoG	LBP-P
AirportHall	0.653	0.496	**0.659**	0.579	0.503
Bootstrap	**0.725**	0.514	0.649	0.541	0.520
Curtain	**0.794**	0.775	0.707	0.505	0.714
Escalator	**0.566**	0.445	0.439	0.366	0.539
Fountain	**0.801**	0.425	0.504	0.779	0.753
ShoppingMall	0.648	0.522	0.513	**0.670**	0.629
Lobby	**0.708**	0.029	0.690	0.684	0.523
Trees	**0.611**	0.345	0.222	0.554	0.606
WaterSurface	0.612	0.801	0.570	0.635	**0.822**
Average	**0.680**	0.483	0.550	0.590	0.623
Variance	**0.007**	0.052	0.024	0.014	0.013

Table 2. F1 results on the Star dataset

6. CONCLUSION

We propose a method for change detection in video. The background model is represented by samples of local ternary pattern and photometric features. We propose new local ternary patterns which are generated from the same color channels as well as from different color channels. The local ternary patterns make full use of all color channels. The features derived from the patterns are more informative and distinctive than other pattern-based methods that use gray values or the same color channel. In the change detection process, a current pixel is classified as background only when both pattern and photometric features match with one background sample. Otherwise, that pixel is classified as foreground (change region). Features of background pixel will be used to update the background model. Finally, we propose a probabilistic refinement to improve each change region by taking into account the spatially consistency of image features. We compare our method with various background modeling algorithms on two video datasets. Our method can achieve better and more consistent performance than all other methods.

7. REFERENCES

[Bar09a] Barnich, O., and Van Droogenbroeck, M. ViBe: a powerful random technique to estimate the background in video sequences. Proceedings of International Conference on Acoustics, Speech and Signal Processing pp.945-948, 2009.

[Bou11a] Bouwmans, T. Recent advanced statistical background modeling for foreground detection: a systematic survey. Recent Patents on Computer Science Vol. 4, No. 3, pp.147-176, 2011.

[Cha16a] Chan, K.L. Background modeling using perception-based local pattern. Proceedings of 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2016, pp.253-260, 2016.

[Elg02a] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of IEEE Vol. 90, No. 7, pp.1151-1163, 2002.

[Gon10a] Gonzalez, R.C., and Woods, R.E. Digital Image Processing. Pearson/Prentice Hall 2010.

[Haq13a] Haque, M., and Murshed, M. Perception-inspired background subtraction. IEEE Transactions on Circuits and Systems for Video Technology Vol. 23, No. 12, pp.2127-2140, 2013.

[Hei06a] Heikkilä, M., and Pietikäinen, M. A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 28, No. 4, pp.657-662, 2006.

 ISBN 978-80-86943-41-1

[Hof12a] Hofmann, M., Tiefenbacher, P., and Rigoll, G. Background segmentation with feedback: the Pixel-Based Adaptive Segmenter. Proceedings of IEEE Workshop on Change Detection at IEEE Conference on Computer Vision and Pattern Recognition pp.38-43, 2012.

[Kim05a] Kim, K., Chalidabhongse, T.H., Harwood, D., and Davis, L.S. Real-time foreground-background segmentation using codebook model. Real-Time Imaging Vol. 11, pp.172-185, 2005.

[Li04a] Li, L., Huang, W., Gu, I.Y.-H., and Tian, Q. Statistical modelling of complex backgrounds for foreground object detection. IEEE Transactions on Image Processing Vol. 13, No. 11, pp.1459-1472, 2004.

[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. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition pp.1301-1306, 2010.

[Ma12a] Ma, F., and Sang, N. Background subtraction based on multi-channel SILTP. Proceedings of Asian Conference on Computer Vision pp.73-84, 2012.

[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 Transactions 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 Transactions on Image Processing Vol. 24, No. 1, pp.359-373, 2015.

[Toy99a] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. Wallflower: principles and practice of background maintenance. Proceedings of International Conference on Computer Vision pp.255-261, 1999.

[Ziv04a] Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. Proceedings of International Conference on Pattern Recognition pp.28-31, 2004.

 ISBN 978-80-86943-41-1

Figure 2. Background subtraction results on the Wallflower dataset (6 image sequences) – original image frames (first column), results obtained by ViBe (second column), results obtained by SILTP (third column), results obtained by MC-SILTP (fourth column), results obtained by our method (fifth column), ground truths (last column).

ISBN 978-80-86943-41-1

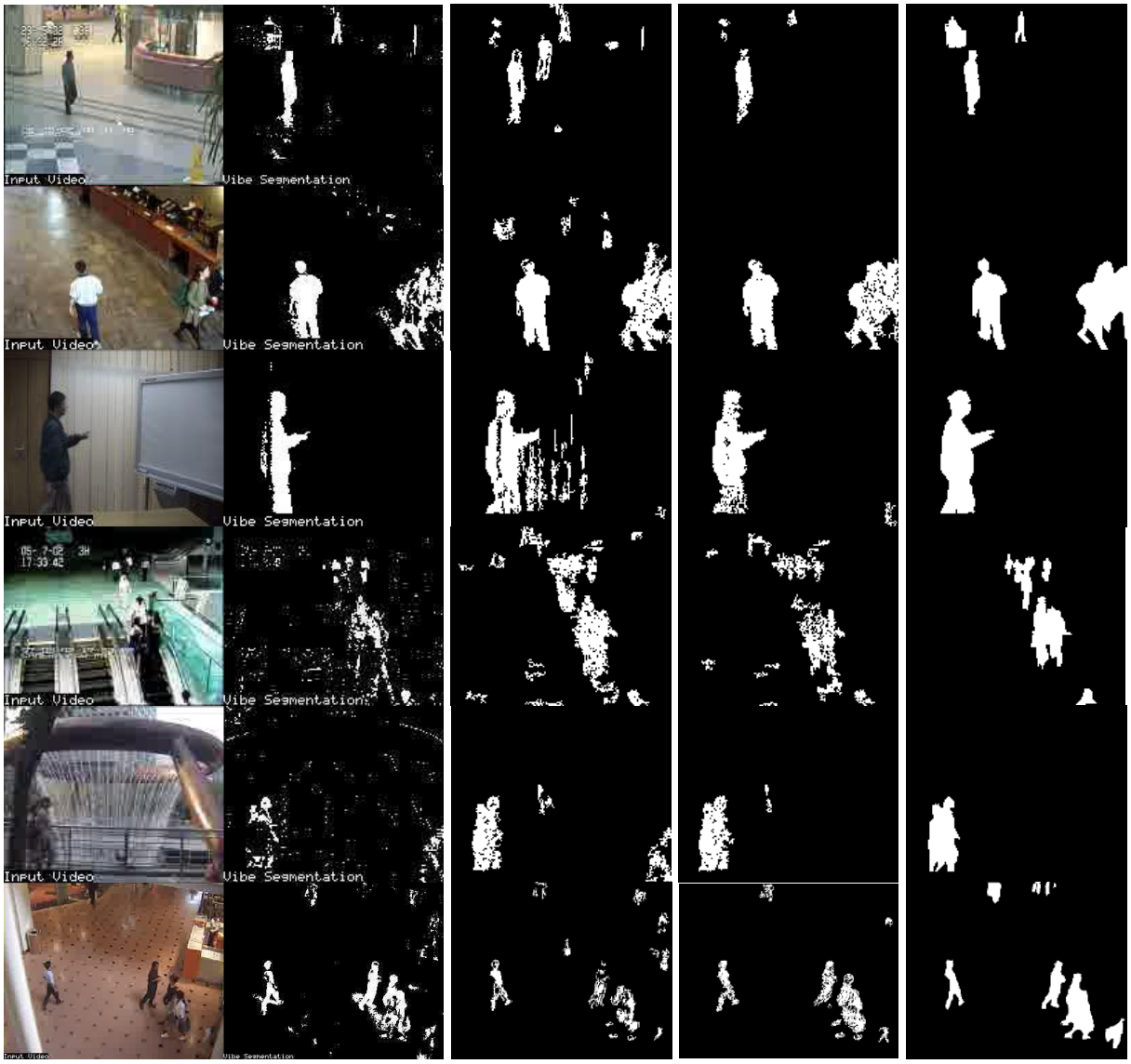


Figure 3. Background subtraction results on the Star dataset (9 image sequences) – original image frames (first column), results obtained by ViBe (second column), results obtained by MC-SILTP (third column), results obtained by our method (fourth column), ground truths (last column).

ISBN 978-80-86943-41-1

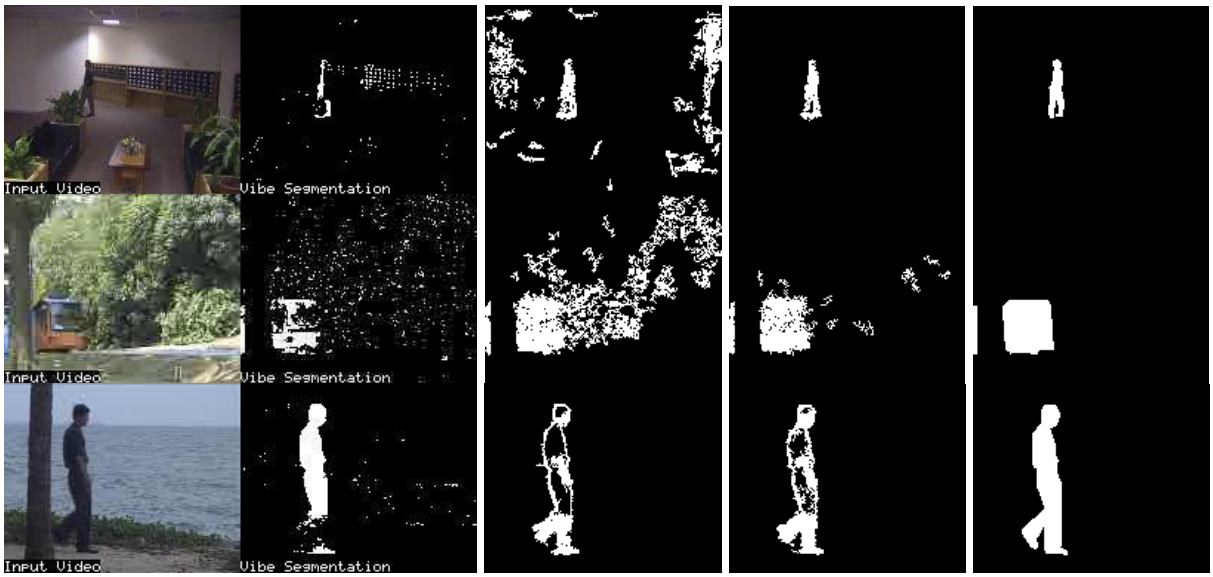


Figure 3. continue.

ISBN 978-80-86943-41-1

Structural Identification of Crystal Lattices Based On Fuzzy Neural Network Approach

Dmitriy Kirsh
Samara University,
IPSI - Branch of the FSRC "Crystallography and Photonics" of RAS
443001, Samara, Russia
kirshdv@gmail.com

Alexandr Kupriyanov
Samara University,
IPSI - Branch of the FSRC "Crystallography and Photonics" of RAS
443001, Samara, Russia
alexkupr@gmail.com

Rustam Paringer
Samara University,
IPSI - Branch of the FSRC "Crystallography and Photonics" of RAS
443001, Samara, Russia
rusparinger@gmail.com

Olga Soldatova
Samara University,
443001, Samara, Russia
op-soldatova@yandex.ru

Ilya Lyozin
Samara University,
443001, Samara, Russia
ilyozin@yandex.ru

Irina Lyozina
Samara University,
443001, Samara, Russia
chuchyck@yandex.ru

ABSTRACT

Each crystal nanostructure consists of a set of minimal building blocks (unit cells) which parameters comprehensively describe the location of atoms or atom groups in a crystal. However, structure recognition is greatly complicated by the ambiguity of unit cell choice. To solve the problem, we propose a new approach to structural identification of crystal lattices based on fuzzy neural networks. The paper deals with the Takagi-Sugeno-Kang model of fuzzy neural networks. Moreover, a three-stage neural network learning process is presented: in the first two stages crystal lattices are grouped in non-overlapping classes, and lattices belonging to overlapping classes are recognized at the third stage. The proposed approach to structural identification of crystal lattices has shown promising results in delimiting adjacent lattice types. The structure identification failure rates decreased to 10 % on average.

Keywords

crystal lattice, fuzzy neural networks, crystal structure identification, lattice system, unit cell, Takagi-Sugeno-Kang neural network, Wang-Mendel neural network.

1. INTRODUCTION

Being the fundamental concept of crystallography and having Angstrom-order sizes, Bravais lattices are building blocks for all crystals. Every crystal is constructed of these lattices in various modifications. At the same time, different crystals can have the same lattices. There is a total of 14 such lattices. Depending on special symmetry, all crystals are distributed among seven lattice systems: triclinic, monoclinic, tetragonal, orthorhombic, trigonal, hexagonal, and cubic systems [Til01a]. Figure 1 presents the general arrangements of Bravais lattices (smallest structural blocks) for each lattice system.

The type of a lattice system is determined by six parameters of a Bravais lattice: the lengths of the three edges and three angles between them (Fig. 2) [Kup01a]. Table 1 shows properties of the lattice systems.

The task of recognizing nano-scale images, which are projections of crystal lattices, can be reduced to the structure identification problem. However, the major difficulty is the ambiguity in choosing a two-dimensional basic cell for a particular projection (Fig. 3) [Ham01a].

Since the classes of Bravais lattices are overlapping, our idea is to use fuzzy neural networks. This kind of networks combines learning and generalization abilities of neural nets, fuzzy logic operations (which allow us to determine the degree of class inclusion of an object as a real number from 0 to 1), and possibility to classify fuzzy rule-oriented bases. A class with the highest degree of class inclusion is the result of structure identification.

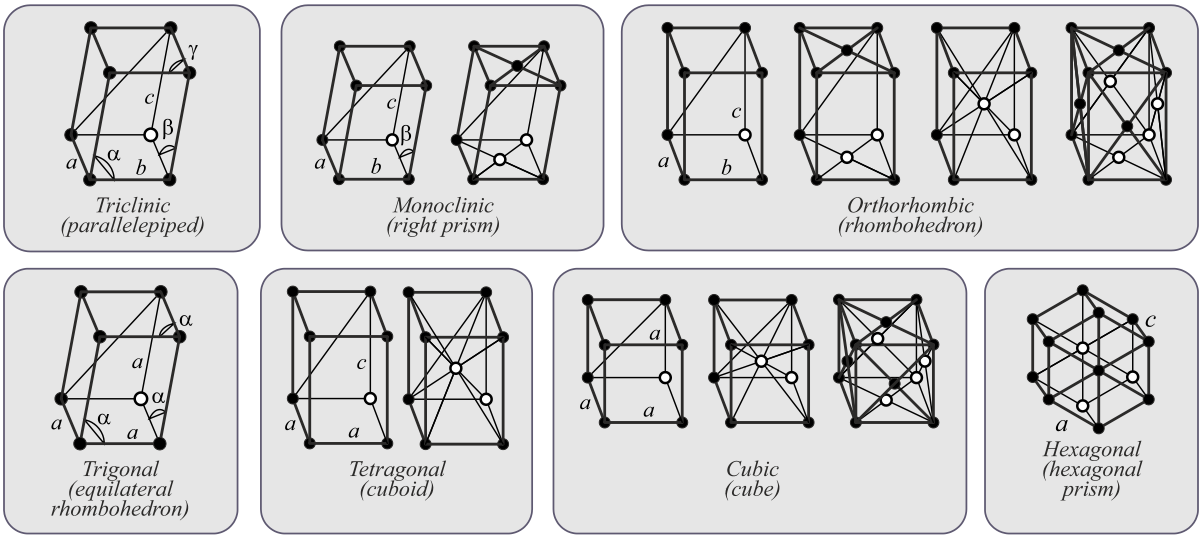


Figure 1. The unit cells of seven lattice systems.

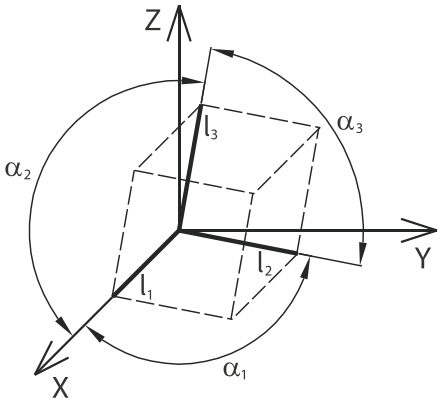


Figure 2. The main parameters of a Bravais unit cell.

Lattice system	Sym.	Edges	Angles
Triclinic	*aP*	$l_1 \neq l_2 \neq l_3$	$\alpha_1 \neq \alpha_2 \neq \alpha_3$
Monoclinic	*mP*	$l_1 \neq l_2 \neq l_3$	$\alpha_1 = \alpha_2 = 90° \neq \alpha_3$
Orthorhombic	*oP*	$l_1 \neq l_2 \neq l_3$	$\alpha_1 = \alpha_2 = \alpha_3 = 90°$
Tetragonal	*tP*	$l_1 = l_2 \neq l_3$	$\alpha_1 = \alpha_2 = \alpha_3 = 90°$
Cubic	*cP*	$l_1 = l_2 = l_3$	$\alpha_1 = \alpha_2 = \alpha_3 = 90°$
Trigonal	*hR*	$l_1 = l_2 = l_3$	$\alpha_1 = \alpha_2 = \alpha_3 \neq 90°$
Hexagonal	*hP*	$l_1 = l_2 \neq l_3$	$\alpha_1 = 120°$; $\alpha_2 = \alpha_3 = 90°$

Table 1. Lattice systems properties.

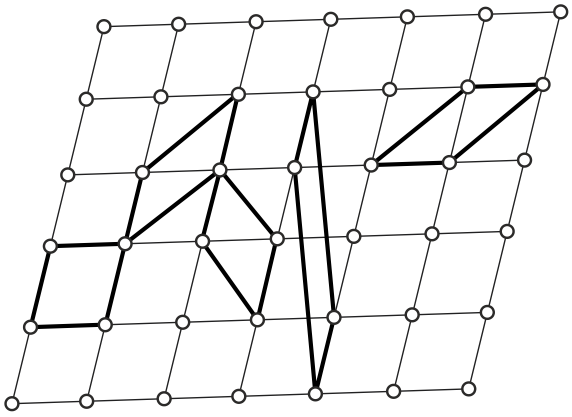


Figure 3. Ambiguity of unit cell choice for a two-dimensional basic cell.

The determination of classification parameters from experimental results and expert evaluation of the parameters are the most popular classification methods using fuzzy neural nets. Particularly, the author of paper [Vin01a] modifies Takagi-Sugeno-Kang (TSK) neural network by introducing the recurrent TSK net. The trick allows the automatic generation of fuzzy rules, but increases the computational complexity of the learning algorithm. Paper [Kip01a] proposes a fuzzy TSK neural network for tackling the classification problem. The net uses the expert evaluation method to choose the most informative classification features and form fuzzy inference rules. In paper [Kat01a] similar approaches are used for learning the author's modification of the Wang-Mendel network. The drawback of the method is the use of subjective estimations of fairly large number of experts and necessity to evaluate their consistency.

Conventional fuzzy rule-based neural net models and modified TSK network use the algebraic product or

 ISBN 978-80-86943-41-1

minimum-form logical product as a fuzzy Boolean conjunction. Respectively, these models use algebraic sum or maximum-form Boolean sum as a fuzzy Boolean disjunction [Kat01a, Oso01a, Rut01a, Vin01a]. At the same time research [Nov01a] allows a conclusion about the effective use of fuzzy logical operations used in algebras of Goedel, Goguen and Lukasiewicz. Paper [Sol01a] offers and investigates modifications of Wang-Mendel networks that allows us to operate fuzzy logical operations defined in these algebras.

The paper is aimed at solving the crucial problem of ambiguity of unit cell choice that greatly decrease the quality of crystal structure recognition. We propose a new approach to structural identification of crystal lattices based on fuzzy neural network. In particular, the fuzzy TSK neural network model has been investigated using a sample of 7000 parameter sets of Bravais lattices belonging to 7 lattice system classes.

This paper is organized in the following way. At first, existing parametric identification approaches are described. Afterwards, we will explain the proposed fuzzy network model and learning technique. The last two sections are devoted to the identification method comparisons, error analysis and conclusions.

2. PARAMETRIC IDENTIFICATION APPROACHES

One of possible approaches to the determination of crystal lattice type is offered in [Kup01a] where previously estimated lattice parameters are compared with predefined reference lattice parameters. The lattice is considered to belong to a particular type if its parameters have the closest match with the parameters of the reference lattice of this type.

Among basic lattice structure identification methods based on parameter estimation are:

- the comparator of the National Institute of Standards and Technology [Kes01a],
- packing efficiency-based identification (Fig. 4) [Smi01a],
- isosurface-based identification (Fig. 5) [Pat01a].

However, these approaches have some drawbacks that restrict their use: the tricky process of crystal preparation (the need for accurate polishing and mounting), low efficiency of comparison of similar lattices, high sensitivity to minor distortions of lattice node coordinates.

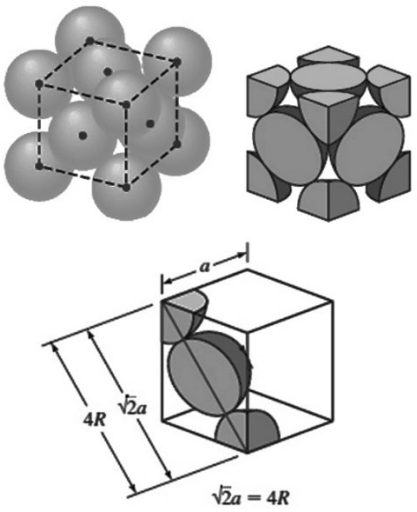


Figure 4. Close packing of spheres.

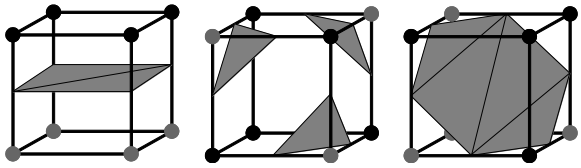


Figure 5. Types of isosurfaces constructed for a cubic lattice.

To overcome these drawbacks, we proposed a new algorithm for crystal lattice parametric identification based on the gradient steepest descent method [Shi01a]. In the algorithms, the result vectors of the lattice identification method based on estimation of Bravais unit cell parameters was used as the initial approximation. The main idea was to increase identification accuracy by the successive refinement of initial estimations (Fig. 6).

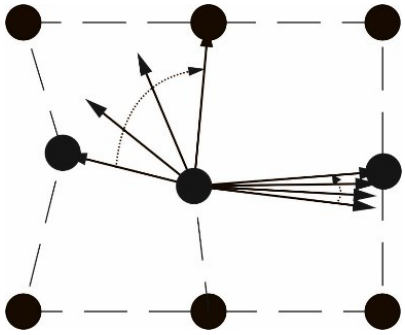


Figure 6. Refinement of translation vectors.

The proposed algorithm showed surprisingly high accuracy of parametric identification at the expense of high computational complexity. Nevertheless, the algorithm did not solve the problem of ambiguity of unit cell choice. In this paper, we offer a radically new approach based on fuzzy neural networks to solve the main problem of crystal lattice structural identification.

ISBN 978-80-86943-41-1

3. MODEL OF FUZZY NEURAL NETWORKS

Figure 7 shows an example of fuzzy TSK multiple-output neural network.

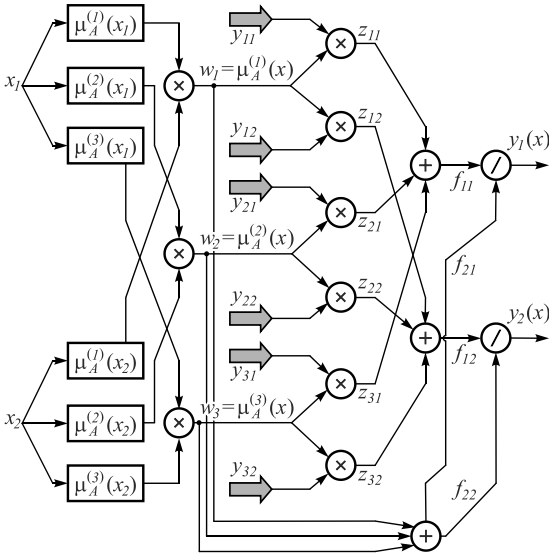


Figure 7. The structure of fuzzy TSK neural network with two inputs, three inference rules and two outputs.

Generalized Gauss function

$$\mu_A\left(x_j\right)=\frac{1}{1+\left(\frac{x_j-c_j}{\sigma_j}\right)^{2b_j}}. \qquad (1)$$

is used as a fuzzification function for each variable x_j .

The fuzzy conjunction in the form of algebraic product

$$\mu_A^{(i)}\left(x\right)=\prod_{j=1}^{N}\left[1\Big/\left(1+\left(\frac{x_j-c_j^{(i)}}{\sigma_j^{(i)}}\right)^{2b_j^{(i)}}\right)\right] \qquad (2)$$

is used to aggregate the condition of the *i*-th rule.

Given *M* inference rules, the aggregation of the network output is done by Equation 3, which can be represented as

$$y\left(x\right)=\frac{1}{\sum_{i=1}^{M}w_i}\sum_{i=1}^{M}w_i y_i\left(x\right) \qquad (3)$$

where $y_i\left(x\right)=p_{i0}+\sum_{j=1}^{N}p_{ij}x_j$ is the aggregation of implication. Weights w_i in this expression are interpreted as components $\mu_A^{(i)}\left(x\right)$ defined by Equation 2.

The first layer of the network is responsible for fuzzification of each variable $x_j\left(j=1,2,...,N\right)$ defining the coefficient of belonging $\mu_A^{(i)}\left(x_j\right)$ for each *i*-th inference rule according to the fuzzification function used. This is a parametric layer whose parameters $\left(c_j^{(i)},\sigma_j^{(i)},b_j^{(i)}\right)$ are subject to adaptation in learning.

The second layer makes aggregation of particular variables x_j defining the resulting coefficient of belonging $w_i=\mu_A^{(i)}\left(x\right)$ in accordance with Equation 2. The third layer is the TSK function generator that calculates $y_i\left(x\right)=p_{i0}+\sum_{j=1}^{N}p_{ij}x_j$. In addition, this layer computes the products of signals $y_i\left(x\right)$ and weights w_i found in the previous layer. This is a parametric layer with adaptable linear weights p_{ij} $(i=1,2,...,M;j=1,2,...,N)$.

The forth layer has two neuron-adders, one of which calculates the weighed sum of signals $y_i\left(x\right)$, and the other sums up the weights $\sum_{i=1}^{M}w_i$.

The fifth layer consists of several output neurons. This is a normalizing layer where the weights are normalized according to Equation 3. Output signals $y_s\left(x\right)$ are defined as

$$y_s\left(x\right)=f_s\left(x\right)=\frac{f_{1s}}{f_{2s}} \qquad (4)$$

4. THE LEARNING TECHNIQUE

The data of generated unit cells of 7 different types were used for learning the neural network. The data were generated under the following conditions:

1. The number of lattices per each lattice system is 1000.

2. The minimum admissible difference between "unequal" cell edges is 0.050 angst.

3. The minimum admissible difference between "unequal" cell angles is 0.020 rad.

4. The maximum admissible difference between the reference and estimated values of cell edges is 0.010 angst.

5. The maximum admissible difference between the reference and estimated values of cell angles is 0.010 rad.

The parameters of unit cell generation are:

ISBN 978-80-86943-41-1

1. The minimum edge lengths are 1.000 angst, 1.000 angst, 1.000 angst.

2. The maximum edge lengths are 5.000 angst, 5.000 angst, 5.000 angst.

3. The minimum angle values 0.175 rad, 0.175 rad, 0.175 rad.

4. The maximum angle values 1.571 rad, 1.571 rad, 1.571 rad.

The size of lattice in each direction was taken equal to three nodes. The G6-space notation [And01a] was used to bring the parameters of unit cells to a common value range.

The preliminary examination of original data allowed us to divide 7 lattice types in 4 groups according to the quantity and ordinal numbers of non-zero columns in data files. The grouping of crystal lattices is given in Table 2.

Lattice System Type	l_1^2	l_2^2	l_3^2	$2l_2l_3\cos\alpha_1$	$2l_1l_3\cos\alpha_2$	$2l_1l_2\cos\alpha_3$	Subgroup No.
Triclinic (*aP*)	x	x	x	x	x	x	**1**
Trigonal (*hR*)	x	x	x	x	x	x	**1**
Hexagonal (*hP*)	x	x	x	x	0	0	**2**
Monoclinic (*mP*)	x	x	x	0	0	x	**3**
Orthorhombic (*oP*)	x	x	x	0	0	0	**4**
Tetragonal (*tP*)	x	x	x	0	0	0	**4**
Cubic (*cP*)	x	x	x	0	0	0	**4**

Table 2. Grouping of lattice system types.

After that the TSK neural network was subjected to learning and tested in three stages:

1. Pair training and testing of the neural network for recognition of 2 lattice types;

2. Training and testing of the neural network for recognition of all 7 lattice types;

3. Training and testing of the neural network for recognition of lattice types in subgroups 1 and 4.

5. DETERMINING THE CRYSTAL LATTICE TYPE

The relative error of structure identification in all experiments was calculated as a percentage of identification failures over the whole test lattice collection. At the first stage 6-dimensional vectors comprising of learning data of two types were fed to the TSK neural net. The output layer held two neurons according to the number of classes being recognized. The results are shown in Table 3.

It is worth noticing that the neural net could not discriminate triclinic lattices (in fact, arbitrary lattices) from trigonal lattices (three equal edges and three equal angles). The reason is that the placing of these two lattice types in a single subgroup is not entirely correct: triclinic lattices are described by six independent parameters (six non-zero columns), and trigonal lattices by two independent parameters (also six non-zero columns). So, we put these two lattice types in one subgroup “formally” rather than “physically”.

At the second stage of the investigation, the data collection presenting all the seven lattice types was used to train the neural net. Six-dimensional vectors made up of this data were fed to the TSK neural net. According to the number of classes to be recognized, the output layer had seven neurons. The experimental results show that the network recognize hexagonal- and monoclinic-type lattices (subgroups 2 and 3) without failure. It is because the learning data for these lattice types has different combinations of zero and non-zero columns than that for other lattice types. In other words, the neural net recognize the lattices of hexagonal and monoclinic type as non-overlapping classes.

	hR	***hP***	***mP***	***oP***	***tP***	***cP***
aP	10	0	0	0	1	1
hR		0	0	0	0	2
hP			15	15	43	12
mP				42	16	10
oP					16	8
tP						12

Table 3. Relative errors of crystal lattice structure identification in pair learning of the TSK network using a 7000-lattice sample.

Additionally, the third stage of experiments was carried out to recognize lattice types belonging to

 ISBN 978-80-86943-41-1

subgroups 1 and 4. The TSK neural net with 6 inputs and 2 outputs were used to deal with lattices of subgroup 1. The same net with 3 inputs corresponding to non-zero columns of initial data and 3 outputs were engaged to process subgroup 4. The identification failure rate of the TSK neural net was 10% for subgroup 1, and 25% for subgroup 4.

Let us compare the values of the relative errors with the results presented in [Kir01a, Kup01a] where the recognition of lattice types was done with the aid of parametric identification methods. By way of example let us look at the best result of structure identification obtained in comparative estimation of Bravais cell parameters and Wigner-Seitz cell volumes [Kup01a] (see Table 4).

	hR	*hP*	*mP*	*oP*	*tP*	*cP*
aP	0	0	1	0	0	0
hR		0	0	2	3	26
hP			7	0	0	0
mP				22	10	0
oP					34	15
tP						26

Table 4. Relative errors of crystal lattice structure identification using parametric identification methods.

The comparison shows that the use of neural nets makes it possible to significantly decrease the structure identification failure rates for the following lattice types:

- trigonal and cubic types from 26 to 2%;

- orthorhombic and tetragonal types from 34 to 16%;

- tetragonal and cubic types from 26 to 12%;

- orthorhombic and cubic types from 15 to 8%.

On the other hand, when discriminating monoclinic and hexagonal lattices from lattices of subgroups 3 and 4, the neural net gives much worse results than parametric identification methods. Particularly, in separation of hexagonal lattices from tetragonal ones the relative error has grown from 0 to 43%.

As for subgroup 4, here the low results are due to the geometric overlapping of classes. A set of cubic-type lattices (red diagonal in Figure 8) lie in the same line in the three-dimensional space. This line is in the plane containing tetragonal-type elements (the dark-grey layer in Figure 8). The plane lies in turn inside the parallelogram formed by orthorhombic-type elements (the light-grey cube in Figure 8).

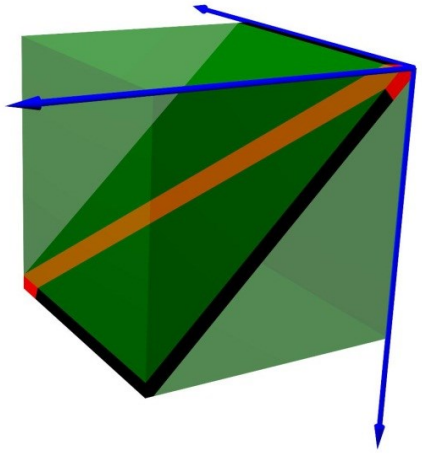


Figure 8. The class overlapping of lattice types of subgroup 4.

6. CONCLUSION

We have offered a three-stage learning technique for neural networks. Crystal lattices are divided into non-overlapping classes in the first two stages. Crystal lattices belonging to overlapping classes are recognized at the last stage.

As compared with parametric identification methods, the use of neural nets makes it possible to decrease the 3D structure identification failure rate for four couples of lattice systems considerably (as much as 2 to 13 times).

The research results allow us to draw a conclusion that fuzzy neural networks are an efficient tool in recognition of crystal lattice types using Bravais cells parameters.

7. ACKNOWLEDGEMENTS

This work was partially supported by the Ministry of education and science of the Russian Federation in the framework of the implementation of the Program of increasing the competitiveness of Samara University among the world's leading scientific and educational centers for 2013-2020 years; by the Russian Foundation for Basic Research grants (# 15-29-03823, # 16-41-630761, # 17-01-00972, # 18-37-00418); in the framework of the state task #0026-2018-0102 "Optoinformation technologies for obtaining and processing hyperspectral data".

8. REFERENCES

[And01a] Andrews, L.C., Bernstein, H.J., Lattices and reduced cells as points in 6-space and selection of Bravais lattice type by projections. Foundations of crystallography, 44(6), pp.1009-1018, 1988

[Ham01a] Hammond, C., The basic of crystallography and diffraction, 3rd ed. Oxford University Press, pp.55-83, 2009

 ISBN 978-80-86943-41-1

[Kat01a] Katasyov, S.A., Software for Building Fuzzy Rule Inference Data Bases for Expert Systems. Fundamental Research, 10(9), pp.1922-1927, 2013

[Kes01a] Kessler, E.G., Henins, A., Deslattes, R.D., Nielsen, L., Arif M., Precision comparison of the lattice parameters of silicon monocrystals. Journal of research of the national institute of standards and technology. 99(1), pp.1-18, 1994

[Kip01a] Kiper, A.V., Stankevich, T.S., Development of a Fuzzy Classifier Using the Sugeno Fuzzy System for Ranking a Fire on the Seaport Premises. Astrakhan State Technical University Bulletin, Marine Engineering Series, 2, pp.18-25, 2012

[Kir01a] Kirsh, D.V., Kupriyanov, A.V., Parallel implementations of parametric identification algorithms for three-dimensional crystal lattices. CEUR Workshop Proceedings, 1638, pp.451-459, 2016

[Kup01a] Kupriyanov, A.V., Kirsh, D.V., Estimation of the Crystal Lattice Similarity Measure by Three-Dimensional Coordinates of Lattice Nodes. Optical Memory & Neural Networks (Information Optics), 24(2), pp. 145-151, 2015

[Nov01a] Novak, V., Perfilieva, I., Mockor, J., Mathematical principles of fuzzy logic. The Kluwer International Series in Engineering and Computer Science, 517, 352 pages, 1999

[Oso01a] Osovsky, S., Data Processing Neural Nets. Finance and Statistics, 344 pages, 2002

[Pat01a] Patera, J., Skala, V., Centered cubic lattice method comparison. Proceedings of algoritmy, pp.309-319, 2005

[Rut01a] Rutkovskaya, D., Pilinsky, M., Rutkovsky, L., Neural Nets, Genetic Algorithms and Fuzzy Systems. Hotline-Telecom, 52 pages, 2006

[Shi01a] Shirokanev, A.S., Kirsh, D.V., Kupriyanov, A.V., Development of the crystal lattice parameter identification method based on the gradient steepest descent method. 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2016, CSRN 2603, pp. 65-68, 2016

[Smi01a] Smith, W.F., Foundations of Materials Science and Engineering. McGraw-Hill, pp.67-107, 2004

[Sol01a] Soldatova, O.P., Lyozin, I.A., Solving the Classification Problem Using Mamdani-Zadeh System-Based Fuzzy Neural Rule Inference Networks. Samara Research Center Bulletin. Physical and Mathematical Sciences Series, 16(2), pp.136-148, 2014

[Til01a] Tilley, R., Crystals and crystal structure. John Wiley & Sons, pp.17-32, 2006

[Vin01a] Vineetha, S., Bhat, C.C.S., Idicula, S.M., MicroRNA–mRNA interaction network using TSK-type recurrent neural fuzzy network. Gene, 515(2), pp.385-390, 2013

 ISBN 978-80-86943-41-1

AgeRegression: Rejuvenating 3D-Facial Scans

Katharina Legde
BTU
Cottbus-Senftenberg
Platz der Deutschen
Einheit 1,
03046 Cottbus, Germany
Legdekat@b-tu.de

Susana Castillo
BTU
Cottbus-Senftenberg
Platz der Deutschen
Einheit 1,
03046 Cottbus, Germany
castillo@b-tu.de

Douglas W. Cunningham
BTU
Cottbus-Senftenberg
Platz der Deutschen
Einheit 1,
03046 Cottbus, Germany
cunningham@b-tu.de

ABSTRACT

The majority of virtual agents have adult bodies. There are, however, a number of reasons for using younger avatars. For example, an adult interface agent usually leads users to expect adult-level communicational and social skills. As a result, users tend to be rather intolerant when the interface agent makes obvious mistakes (e.g., incorrect grammar) or uses inappropriate behavior (e.g., looking away from the interlocutor). Since computer social-skills are still under-developed, it seems reasonable to use a body model that reflects this: child avatars. Unfortunately, the use of database-driven techniques for creating a variable-aged animation system would require a very large number of scans of children at different ages, making such a system impractical for technical and ethical reasons. As an alternative, this paper develops and validates a method for synthetically and systematically altering the apparent age of a virtual character. The here proposed technique is able to create younger and older versions of a facial scan and guarantees that the resulting meshes can be animated. Starting with a three-dimensional, adult facial scan, we use a physiologically-inspired, trigonometric polynomial to age-regress the model to a desired age. Quantitative measurements show that the technique can reconstruct the correct anthropometric proportions of 2-10 year-old children. A perceptual experiment provides an initial mapping of the technique's parameters onto the perceived age and realism.

Keywords

Age Regression, Cardioidal Transformation, Face Models

1 INTRODUCTION

Human faces convey an impressive amount of information. We intuitively and automatically detect static facial cues like gender, identity, mental state, health, and age [Cah90, Ram09, Fis16] and use them to determine how to interact with an interlocutor (e.g., [Rya86, Gle75, Yar09]). Synthesizing the human face is especially challenging. Still, there are a wide range of applications using an artificial human face including advertisement, entertainment, education and user interfaces. User interfaces are supposed to ease the communication in between the user and the machine with the help of e.g. an intuitive graphical user interface, speech recognition or with a virtual agent. Virtual agents are granting the computer human-like communication and social abilities (for examples see [Kop05, Kro17, Mor15, Nie09]). In most cases they are represented with human adult bodies. Especially when the computer "looks" like an adult we expect it to also act like an adult, including the use of appropriate social behavior, proper grammar, intonation, and body language [Vin06, McD12]. We also expect the virtual agent to understand what we say at an adult level. These expectations are often not met. This conflict between appearance and behavior usually leads – at the very least – to a frustrated and confused user [Vin06, McD12] who no longer wishes to interact with the system.

This paper combines the suggestion that the appearance of a virtual agent should match its behavior, the underlying task, and the system constraints [Kru12] with the observation that we adjust our behavior (including what we say and how we say it) and our expectations to the other person's age [Gle75, Rya86, Yar09]. Since people tend to be very tolerant to social and communication mistakes made by children as well as to their constrained level of general knowledge we propose that

virtual agents that look like children can improve the interactions with computers.

Creating a virtual child poses a number of technical and ethical difficulties. Among the core concerns is that modern performance-driven animation techniques require a large number of three-dimensional (3D) facial scans and motion capture information (e.g., for different expressions). Furthermore, if the interface is to be able to provide a range of apparent ages, then scans and recordings from many ages will have to be made. Long recording sessions of carefully presented facial expressions is already difficult with adults. With small children, such recording sessions are nearly impossible. These technical concerns (and the related ethical concerns of using a real-child's face as an interface agent) can largely be side-stepped by creating and animating an adult face, and then rejuvenating it to the desired age.

The age-rejuvenation technique proposed in this paper uses a trigonometric polynomial inspired by Todd et al.'s revised Cardioidal Transformation model [Tod80]. Unlike the original technique, the proposed polynomial works on the 3D facial structure, allows different local facial areas to grow at different rates, and can be used for age-regression as well as age-progression. After describing the technique, we show a range of possible results and then map the parameters of the polynomial to perceived age (and realism) through a perceptual experiment. Finally, we then compare the anatomical proportions of age-regressed faces to those found in real child pictures of the regressed individual.

2 AGE SYNTHESIS

Age synthesis describes a re-rendering of a face image or a face mesh with an approximation of natural aging or rejuvenating effects.

2.1 Natural aging

The process of natural aging causes significant, idiosyncratic changes in a person's face. The appearance of a person's face at any given age is highly dependent on both inner and outer factors [Enl89, Alb11]. Inner factors are mostly biological. For example, during aging soft tissue looses its elasticity and volume, fatty tissue gets lost, and the face increasingly resembles the bony structure of the skull [Par08, Alb11]. Other inner factors include biological sex and ethnic group [Alb11]. Biological sex determines the intensity and duration of the actual aging process and influences the growth of facial bones, the height of cheekbones, the width of the nose and the existence of facial hair [Enl89]. Outer factors, on the other hand, generally describe changes arising from environmental causes or lifestyle-based behaviors and refer to things such as weight, scars, wrinkles due to frequently used facial expressions, effects of the physical environment (such as frequent exposure to the sun), and the general life experiences of the person [Alb11]. The particular combination of inner and outer factors that occur for a given individual contribute strongly to the individuality of each human face and must be considered when age synthesis is preformed.

Aging is not a linear process. The changes that occur to adults are fundamentally different from those that happen during childhood [Suo07]. During childhood, skin changes are minor while the cranium undergoes various and rapid growth modifications. In adult aging, in contrast, the shape of the cranium changes very little while skin changes can be considerable, including changes due to sunlight and gravity [Pit75, Suo07, Alb11].

2.2 Age synthesis techniques

Age synthesis can be classified into the three groups: explicit mechanical, implicit statistical, and explicit data-driven [Fu10].

Explicit mechanical synthesis: Explicit mechanical synthesis techniques describe changes of skin and other facial tissue during growth, including the synthesis of wrinkles [Sha04, Fu10]. Wu et al. presented a plastic-viscoelastic model, which simulates dynamic wrinkles [Wu94]. In their technique, the face is represented with three different layers: muscles, connective tissue, and skin. During the contraction of the muscles, the connective tissue – which is modeled with hookian springs – simulates the elastic process and thereby pulls the skin in a specific direction. The main use of this model is to create wrinkles for facial expressions, but Wu et al. [Wu94] also state that changing certain parameters such as the stiffness of the spring and the thickness of the connective tissue will result in wrinkles caused by aging.

Implicit statistical synthesis: By far the most common class of technique is implicit statistical synthesis. This approach does not use physics, but instead tries to capture age-related changes by performing an automated, statistical evaluation of the appearance variations of facial images or 3D facial scans. These techniques require the acquisition of a large training dataset at a wide range of different ages. Each image or mesh in this training set is considered to be a high-dimensional point. Scherbaum et al. [Sch07] used a Morphable Model to extract shape and texture variations in faces in such a dataset. With the help of a non-linear Support Vector Regression performed on several shape and texture coefficients, they are able to learn a function which maps every face in their database to a scalar age value. Calculating the gradient of the age function allows Scherbaum et al. [Sch07] to extract aging trajectories for shape and texture and thereby alter the apparent age of 3D facial scans.

 ISBN 978-80-86943-41-1

Another representative approach is the Multi-Resolution Dynamic Model presented by Sou et al. [Suo07]. Given an image as input, they first build a graph structure and then sample it over various age groups according to a beforehand learned dynamic model through a Markov process. This allows them not only to capture and afterwards simulate aging effects relatively well, but also to simulate *variations* in the aging process [Suo07].

Golovinskiy et al. [Gol06] proposed a statistical face model to extract and transfer facial details, like wrinkles or pores. They use high resolution facial scans and split them into a smooth base mesh and a detailed displacement image. Statistics are performed to capture the local orientations and the amount of detail. These statistics can now be used in combination with a displacement image of another facial scan. Thereby they are able to transfer wrinkles of one mesh to another, to make the mesh seem older or smooth the wrinkles, to make the mesh seem younger.

Explicit, data-driven synthesis: Explicit, data-driven synthesis techniques use empirical, mathematical models of human growth to describe the shape changes of the head. Such techniques represent a human face as a 3D-mesh and then alter the mesh with geometrical scaling, translation, and rotation operations. These techniques use explicit models of anatomical changes of the aging process. One of the most well-known models of age progression is the **Craniofacial Growth Model** [Tho17], which simulates the aging process by modeling shape changes caused by craniofacial growth. The model is based on the observational work of D'Arcy Thompson, who believed that the morphological changes of the aging process can be reduced to simple geometrical transformations [Tho17, Tod80]. Pittenger and Shaw asserted that growth was a series of visco-elastic events [Pit75]. They observed that during aging the face grows more rapidly than the rest of the cranium which results in changes in the face shape profiles, the so called the facial angle [Tod80]. Todd et al. mathematically modeled those events with affine shear and cardioidal strain transformations (CST) [Tod80, Fu10]. With the help of experiments, they found that affine shear transformations have less effect on the perceived age of the facial profiles and produce unidentifiable distortions. They subsequently focused on the cardioidal strain transformation, which had a much larger effect on the perceived age [Tod80]. Specifically, they conceptualized the facial profile as a cross-section of a head that is modeled as a sphere filled with fluid. A transformation on a certain point depends on the force which is exerted on that point, which is derived from a function considering gravity, the density of the fluid and the radius of the sphere [Tod80, Ram06]. They assumed that the pressure is directed radially outwards, distributed continuously and symmetrically along the vertical axis [Tod80, Ram06]. Todd et al. revised that approach with the observation that the amount of pressure at any point is depending on the amount of fluid above it and therefore added a term of position to the original CST [Tod80]. The revised cardioidal strain transformation (rCST) was defined mathematically – for 2D facial profiles – in polar coordinates for (R,θ) with the following equation [Tod80]:

$$R' = R(1 + k \cdot (1 - cos(\theta)) \tag{1}$$

with R being the distance of a given point to the origin before transformation and R' being the distance of that point to the origin after transformation. The polar angle θ is the angle between the line segment R and the vertical or polar axis. After the transformation is applied the polar angle stays the same $\theta' = \theta$. The constant k is a growth-related constant which increases with age [Tod80, Ram06], thereby this equation is defined as an age progression. Mark et al. later extended this equation to work with 3D faces with the help of spherical coordinates [Mar83].

3 PROPOSED AGE REGRESSION

The proposed algorithm attempts to rejuvenate three-dimensional facial models using a 3D trigonometric polynomial (Equation 2) inspired by Todd et al.'s revised CST model (Equation 1). Specifically, we converted the equation to spherical coordinates, made it additive, altered the terms that involve the angle θ so that they are always positive and between [0,1], and added a number of terms to allow changes to specific facial areas (i.e., the nose bridge *nb*, the tip of the nose *nt*, and the cheeks *ch*). We also allow changes along the latitude (azimuth or φ) as well as the longitude (inclination or θ) of facial scans. Finally, all terms in the equation now use exponential functions to focus their changes on different areas. By having different exponents for the lateral and the longitudinal cosines, the new terms can – for example – be tightly focused horizontally but loosely focused vertically (such as would be needed to capture nose-related changes). The new equation is:

$$\begin{aligned} R_{yng} = R_{old} \cdot (1 &+ k_h \cdot (\frac{\cos(\theta)+1}{2})^{n_h} \\ &+ k_{nb} \cdot (\frac{\cos(\theta-\alpha_{nb})+1}{2})^{n_{nb}} \cdot |\cos(\frac{\varphi}{2})^{m_{nb}}| \\ &+ k_{nt} \cdot (\frac{\cos(\theta-\alpha_{nt})+1}{2})^{n_{nt}} \cdot |\cos(\frac{\varphi}{2})^{m_{nt}}| \\ &+ k_{ch} \cdot (\frac{\cos(\theta-\alpha_{ch})+1}{2})^{n_{ch}} \cdot |\cos(\frac{\varphi}{2})^{m_{ch}}|) \end{aligned} \tag{2}$$

with as is standard for spherical coordinates $\theta \in [0,\pi]$ and $\varphi \in [0,2\pi]$. The angle θ is between R_{old} and the vertical polar axis while φ defines changes along the orthogonal plane. R_{old} is the distance of one point in the adult facial mesh to its local origin (center of mass).

 ISBN 978-80-86943-41-1

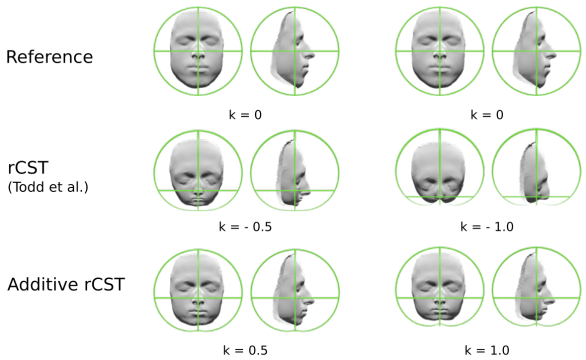


Figure 1: Age Regression equation corresponding to Todd et al. and additive rCST. For illustration purposes the same deformation is applied to the unit circle shown in green.

The age coefficients k_i control how much deformation is applied to distinct areas i. The exponents n_i, m_i control the tightness of the transformation for the distinct areas i. The angles α_i define a shift in the starting point θ of the transformation. The full equation is applied to every vertex of the 3D scan of an adult face.

It should be noted that Todd et al.'s rCST (Equation 1) can already be used to perform an age regression [Mar83]. While small amounts of rejuvenation (using negative values of k close to zero) can produce plausible results, higher rejuvenation effects produce increasing distortions (Figure 1, second row). Note that merely altering the sign of the first term (by adding it and ensuring that the θ terms are in the range [0,1]) already improves the results (Figure 1: third row, only considering the term for the head h in Equation 2). Since the effect of such an additive, positive rCST is still global, it cannot account for the different growth rates of different facial areas (as can be seen, e.g., with the overly-large nose).

To simulate the fact that different facial regions grow at different rates, we added three new terms (affecting the nose bridge, nose tip, and cheeks) based upon anatomical considerations of the differential growth rates (see, e.g., [Ram06]). Initial experiments (see Section 5) confirm that these four terms (the three new ones and the cranium) represent a decent basis model. The angles α_i which assure the focus of the transformation on a certain area are unique for different 3D scans. Note that the angle φ was used in all areas except for the head term h, since its inclusion there leads to unnatural cranial forms. Also note that we use the half angle $0.5 \cdot \varphi$ to ensure that the changes for nb, nt, ch happen only to the face (and not to the back of the head). The aging coefficients $k_h, k_{nb}, k_{nt}, k_{ch}$ are chosen to match biological constraints. During childhood the cranium (the first term in Equation 2) undergoes the most growth modifications [Suo07], thereby the age coefficient k_h should be altered more than, for example, the age coefficient

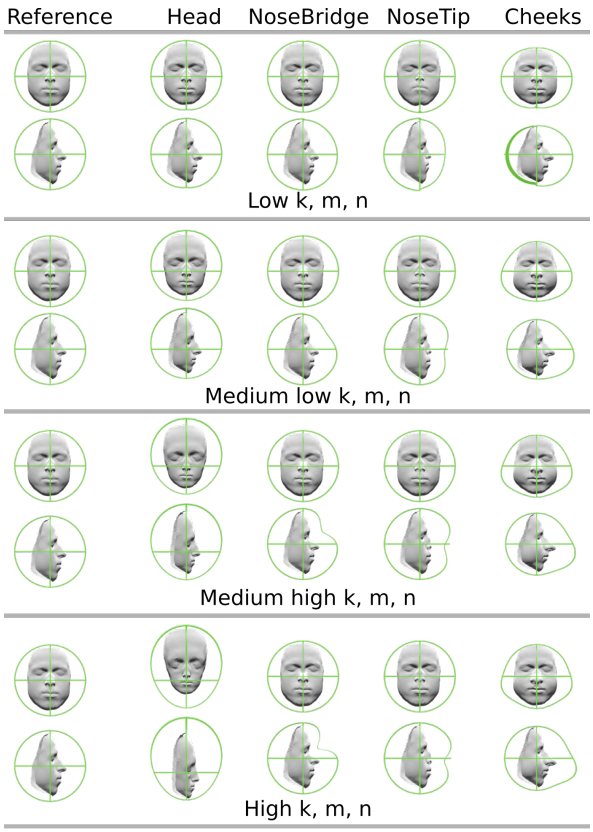


Figure 2: Transformation for different facial areas with varying parameters in absolute value. Only one area (columns) is transformed at a time. For illustration purposes, the same deformation is also applied to the unit circle shown in green.

for the nose bridge k_{nb} or nose tip k_{nt}. We also defined different exponents n, m for each of the four areas to control the tightness of focus of the cosine term. A rejuvenation for the nose tip nt, for example, needs to affect a much smaller area than the transformation of the cheeks ch. For our reference face, the effect of the different parameters for the four terms are illustrated in Figure 2.

4 RESULTS

The proposed technique can produce a very wide range of facial deformations. Careful choice of the parameter values can reduce the apparent age of a 3D adult facial scan to any desired age. Other parameter values, however, can produce unnatural facial meshes. A small sample of possible results considering a subjectively chosen parameter space can be seen in Figure 4. We chose to first linearly interpolate the parameters k_i, n_i and m_i in a proposed interval to rejuvenate the 3D scan of an adult. The faces in the middle of the parameter range clearly resemble children of different ages. Extreme values for k_i, n_i and m_i decrease the degree to which the face appears human-like. Figure 4 also makes it clear that to control the apparent age of

ISBN 978-80-86943-41-1

a face, more than the age coefficient k needs to be considered. In particular, the size of the exponents can play an important role, consistent with Ramanathan et al.'s [Ram06] claim that different areas of the face grow at different rates. A closer glance at the figure shows that, for example, for coefficients $k_h = 0.6$ through $k_h = 0.8$ (sixth through eighth row) with low exponents of n_i and m_i (left side of the spectrum) will produce almost baby-like faces. Higher values for n_i and m_i (right side of the spectrum) will produce older children. It is important to note that the coefficients k_i can also be used for reducing or expanding a region, e.g. using a negative coefficient can help to regulate the nose tip better.

We also applied our method to commercially purchased head scans [Ten24] of different genders and different ethnic groups. Some example results can be seen in Figure 3 for the evaluated age groups and their corresponding parameters from the proportional study (see Section 5.2). Figure 3 shows that we obtain realistic, child-like faces. It is important to note that even for very different facial meshes, the exact same parameter values for k_i,n_i, m_i (see Figure 4) produced similar amounts of age regression. The local origin of both scans is placed in between both eyes in frontal view and in between the eye and the ear from side view (approximately the center of mass of the previous scan). We chose to also use the same angles α_{nb}, α_{nt} and α_{ch} defined for our previous scan to specify the areas in our new scans. Figure 3 shows that we produce with the same angles reasonable child-like faces. Obviously, a manual adjustment of these angles for the new meshes or even an automatic detection would produce even better rejuvenation results. Note here that the inclusion of the neck in the 3D-scan is important and makes a difference in the perception of age (especially the Adam's apple in the male scan). For further results of the full suggested parameter space from Figure 4 for both of the models please see supplemental material.

The rejuvenated meshes can be animated in a number of ways. The simplest method is to age regress a neutral model and use cluster animation based on motion capture data to move the face [Par08]. One can also use blend-shape animation, where a number scans of the same person in different peak expressions are placed into correspondence [Par08]. Novel expressions and animations are then produced by a weighted sum of the different scans. Note that if the correspondence is established before age-regression, then only the neutral expression needs to be regressed since the other scans in a blend-shape animation system are stored as a deformation of the neutral expression. Initial tests of both cluster-based and blend-shape-based animation produce realistic results. Note that in these animations, the motion of an adult was projected on the face of a child. This technique, then will allow future work to determine the relative contribution of dynamic (e.g., motion) and static (e.g., geometry) information to the perception of age.

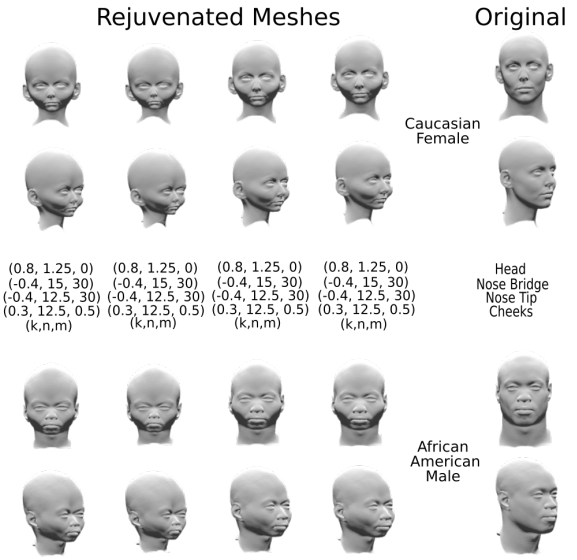


Figure 3: Example results of the proposed technique considering different head models. Values for k_i, n_i, m_i taken from suggested parameter space (see Figure 4). Chosen subset is consistent with proportional study (see Section 5.2).

Additionally, our proposed technique can also be used to make faces older again by simply dividing R by the aging terms instead of multiplying it (see Equation 2). To test this, we first age-regressed an adult face and then progressed it back to the adult face without changing the parameters for k_i, n_i, m_i. We were able to recover reasonable adult faces including the original scan. To use the technique to age-progress an face from young adult through middle-age into old age, an adjustment of the parameter space is needed. During the later stages of aging, the cranium grows less than for example the nose [Ram09], this fact can be modeled by altering the k_i for these areas. Also an adjustment of the k_{ch}, n_{ch}, m_{ch} could be useful to synthesize the loss of fatty tissue.

5 EVALUATION

To evaluate the effectiveness of the technique, and to provide a first approximation of which parameter combinations can yield specific ages, we performed a perceptual experiment and a proportional study using all of the rejuvenated models shown in Figure 4.

5.1 Perceptual Experiment

The perceptual experiment presented a set of stimuli generated by our equation and asked participants to rate the apparent age of the head and how natural or "human-like" the head was.

Methods: Any attempt to systemically test the effect of all 11 parameters of the equation and their combinations would require a prohibitively large number of

ISBN 978-80-86943-41-1

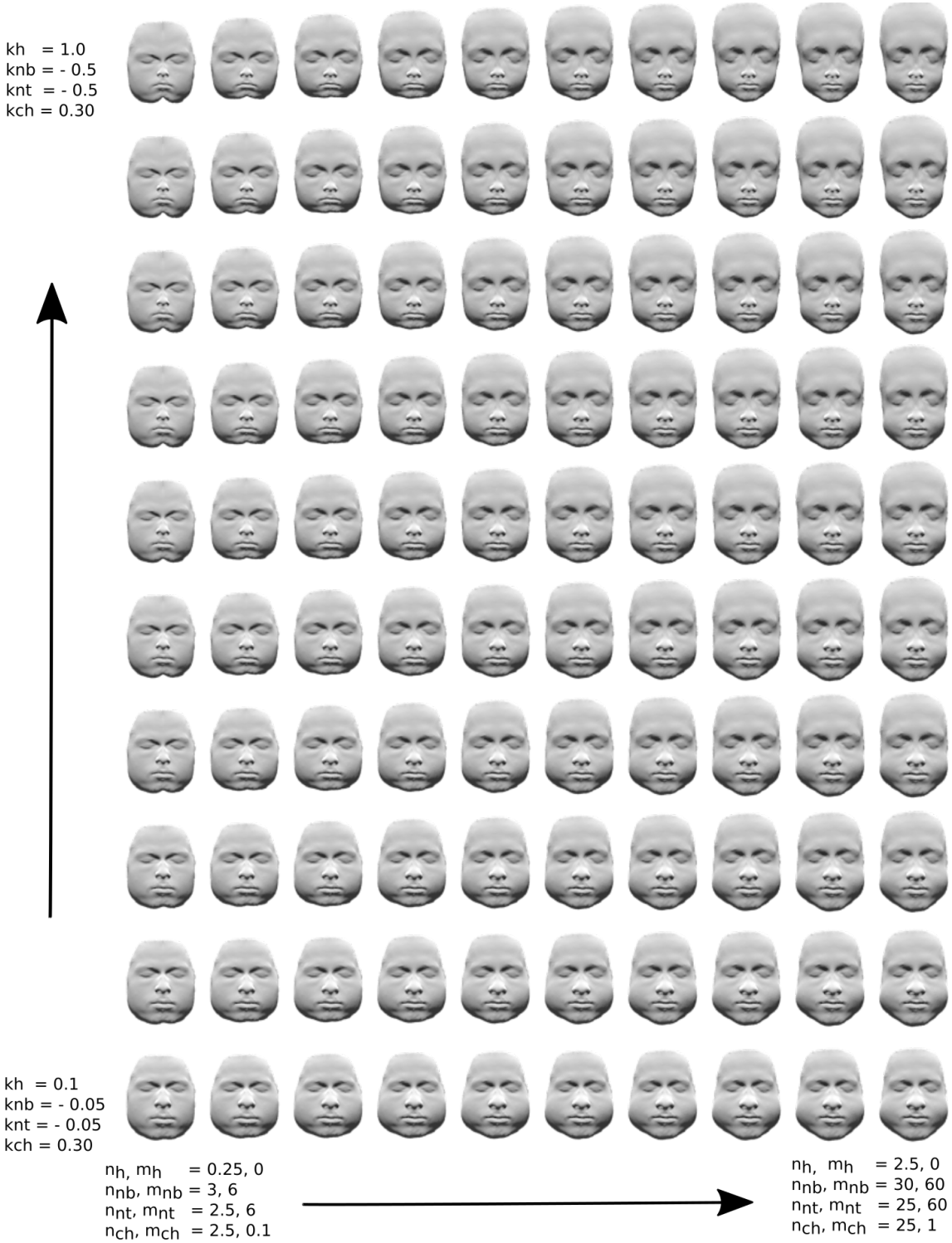


Figure 4: A sample of results of the proposed age regression technique. Parameter values for k_h, k_{nb} and k_{nt} as well as the exponents n, m for the specific areas are given in the picture. For initial testing k_{ch} was held constant with a value of 0.30**.**

ISBN 978-80-86943-41-1

trials. Therefore, we decided to use only a subset of the possible combinations. Specifically, we set the age coefficients k_h,k_{nb} and k_{nt} to different initial values and then linearly modified these values. Note the age coefficient k_{ch} was held constant for the experiment. All age coefficients co-varied perfectly (any change in one was accompanied by an equal change in the other). Each of the exponents n and m also had a different initial value, but again all co-varied perfectly. The combination of 10 head/nose-bridge/nose-tip coefficients with 10 exponents yielded 100 reconstructed faces (see Figure 4). The stimulus set also included the original facial scan. All stimuli were rendered without texture and under the same lighting conditions. On any given trial, one 3D face was shown both a frontal view and with a 60 degree angle to the camera, rendered with orthographic projection in front of a black background.

Ten people (4 women and 6 men; between 24 and 38 years old) participated in the experiment for financial compensation at standard rates. They were naïve to the purpose of the experiment. All participants provided informed consent. Each participant sat alone in front of a computer monitor in a darkened room and was given the experimental instructions. They were then shown all 101 stimuli in random order, with each participant receiving a different order. First, each participant had to rate the apparent age using a set of different age groups: $<1, 1-2, 2-4, 4-6, 6-8, 8-10, >10$ years old. Second, they had to rate how natural or human-like the head was using a 7-point Likert scale (with 7 being extremely likely to be human and 1 being extremely unlikely to be human). To help anchor the naturalness scale, the real 3D scan of the reference person was shown and the participants were told that this was a real person and should be given a value of 7. The average time for one experiment was 30 min.

Results: Overall, the age regression technique significantly and systematically modified the apparent age of a facial scans. Moreover, nearly all of the results were seen as extremely human-like (see Figure 5c). Since the two head coefficients co-varied perfectly as did all of the exponents, there were effectively two factors: age coefficient and exponent. We performed a two-way, repeated measures ANOVA with both of these as within-participants factors. Both main effects and the interaction are significant ($\mathrm{F}(9,81) = 70.23, \mathrm{p} < 0.001$; $\mathrm{F}(9,81) = 6.229, \mathrm{p} < 0.001$; $F(81,729) = 1.63, p < 0.001$ for the age coefficient, exponent, and interaction, respectively). Further analysis showed that the age coefficient and the perceived age are significantly (and negatively) correlated ($r = -0.6918, p < 0.001$; see Figure 5a). If we take a look at the correlation of perceived naturalness and the age coefficient k (see Figure 5b) we see that we produce very natural looking rejuvenated faces (rated above 4 on a 7-point Likert scale) especially in the middle of our spectrum (third till eighth row). As mentioned above, extreme parameter values (bottom and top row) tends to produce somewhat unnatural faces.

Modifying the exponents helped to produce younger faces but also had a distinct effect on the perceived naturalness of the shown faces (see Figure 5d and e). Closer examination shows a small but significant correlation of exponent with the perceived age ($r = -0.1554893, p < 0.001$). As can be seen in Figure 5d, the low correlation is probably due to the fact that their relationship is not linear. Small exponents helped a little to reconstruct younger faces. Larger exponents, however, did not increase the rejuvenation. This effect might be explained with the nose in the profile view of our stimuli set. As the age coefficient of the head k_h mainly stretches the full face, the nose also gets longer and thereby less child-like. As can be seen in Figure 5e, most of the exponents produced natural-looking faces, except for extreme values of our parameter space. In sum, the initial parameter ranges used for the experiment were surprisingly effective, although clearly some fine-tuning on the specific values is needed.

The perceived age was also slightly but significantly correlated with the human-ness of the reconstructed faces (correlation of $r = 0.125098, p < 0.001$). The median (represented as rot dots) and the density of the violins emphasize the fact that almost all of our reconstructed faces were seen as very likely to be human. Note that the distribution of results is non-Gaussian, leading to the assumption that the extreme parameter values are outside the ideal range.

The participants rated the original facial scan older than 10 and on average with 6.4 on the 7-point Likert scale of naturalness.

5.2 Proportional Study

To validate how close we come to the real shape of a child's head we compared pictures of our reference face (who was 28 years old) taken when he was a child (at 2, 4, 7 and 10 years) with our rejuvenated faces considering different anatomical proportion measures. In order to decide which rejuvenated faces best resemble a given real image, we selected the highest rated rejuvenated faces (rated ≥ 5 on a 7-point Likert scale) in terms of naturalness which were most frequently voted for the relevant age group.

Following Chellappa et al.'s and Farkas et al.'s approach [Ram06, Far94] we specify special landmarks in both of the faces, compare their ratios and thereby evaluate if our technique is able to establish the proportions of a specific child's face. The landmarks we chose are a subset of 57 landmarks mentioned by Farkas et al. [Far94] and can be seen in Figure 6 (left-hand side).

The original and the transformed child faces are in point-to-point correspondence. So, we were able to

 ISBN 978-80-86943-41-1

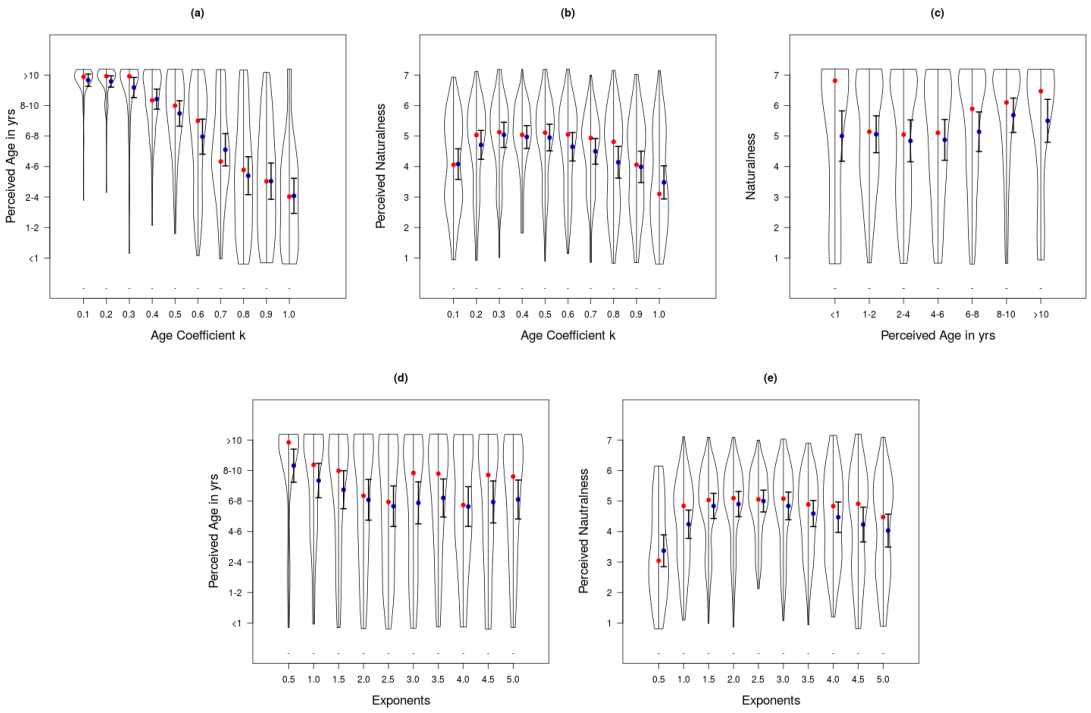


Figure 5: Results of the Experiment: Correlation of (a) perceived age and k, (b) perceived naturalness and k, (c) perceived age and naturalness. (d) perceived age and exponent n, m, (e) naturalness and exponents n, m. Red dots represent the median, blue dots the mean, error bars represent the standard error of the mean, the violins indicate the relative density of the distribution.

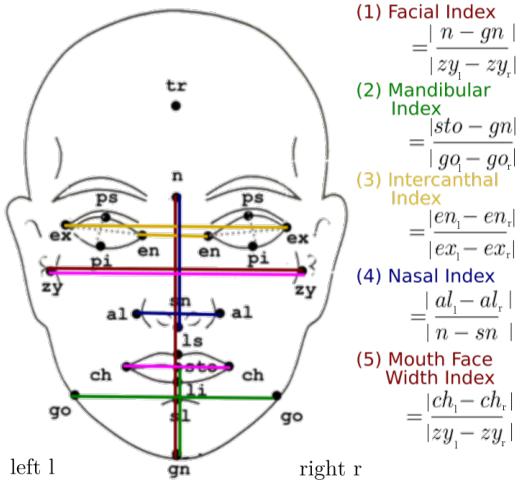


Figure 6: Subset of considered proportions of the face, see [Far94] for the full measurements.

choose exactly the same vertices as landmarks in different rejuvenated models. We then measured the distances among the horizontal and/or vertical axis according to the ratios. To extract the landmarks in the real photos, we manually measured them with the help of a graphics software. Note that low resolution pictures and any out-of-plane rotation (such as turning the head to the side) will artificially alter these "real" values.

In general, our proposed age-regression technique can reconstruct the correct proportions of a specific child. Table 1 shows that facial index (1) values obtained for the rejuvenated faces matched the real images in three out of the four considered age groups. Likewise, the intercanthal index (3) values match very well for every age group. That is a bit surprising since we only modified the eyes as a part of the overall head term and never considered them specifically. The nasal index (4) reveals that our technique works a bit better for older (age ≥ 4) children than younger children. The values for the rejuvenated 2 year old were a bit too low, which might reflect the nose over-stretching mentioned above. The mouth face width index (5) and the mandibular index (2) show a good match for two out of the four reference ages. In particular for index (2) and (5), our values are a little off when it comes to older children.

In Figure 7, the real photos and their corresponding rejuvenated faces with the highest matching ratios among the rated age group are shown. Even though the model and the picture are not always identical, they do have very similar facial proportions (see Table 1) and are clearly perceptually realistic. A finer tuning of the parameters would most likely produce more physically accurate rejuvenated faces.

It is worth mentioning that the age coefficients for 7 and 10 years in Figure 7 are exactly the same. The only difference between these two faces is a small change in

ISBN 978-80-86943-41-1

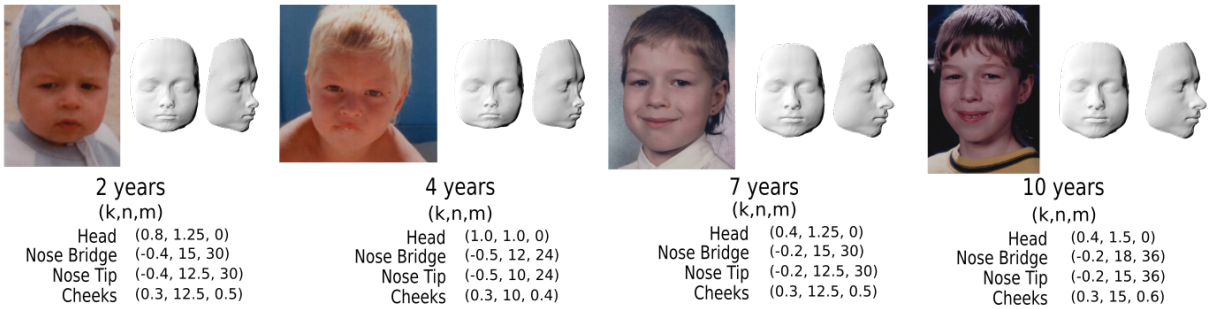


Figure 7: Results for the proportional study: Comparison of the rejuvenated model with original child photographs.

	2 yrs		4yrs		7 yrs		10 yrs	
	real	model	real	model	real	model	real	model
(1)	0.80	0.91	**0.82**	**0.83**	**0.97**	**0.97**	0.96	1.01
(2)	**0.29**	**0.29**	0.38	0.26	0.41	0.31	0.43	0.32
(3)	0.38	0.32	0.36	0.32	0.37	0.31	**0.33**	**0.32**
(4)	0.71	0.59	0.65	0.59	**0.63**	**0.64**	**0.64**	**0.63**
(5)	0.32	0.37	**0.36**	**0.36**	0.52	0.38	0.47	0.38

Table 1: Results of proportional study. Bold numbers show identical ratios. The underlined values show values which are approximately the same (tolerance of 0.06**).**

the exponents n_i and m_i (see Figure 5c). Such a small change in the exponent (e.g. for the k_h a change of 0.5) can be perceived as three years of age difference. We can also see the complex interaction between the age coefficient k_i and the exponents n_i, m_i. Interestingly, in this case the age coefficient was higher (considerably lower for the negatively correlated k_{nb}, k_{nt}) for the four year old than for the two year old. The reduction in age here was caused by the changes in the exponents.

6 CONCLUSIONS AND FUTURE WORK

This paper presents an age-regression algorithm for rejuvenating facial scans. The algorithm takes any 3D polygonal mesh of an adult and applies localized, 3D transformations using a trigonometrical polynomial in order to generate realistic, younger versions of that person. A perceptual experiment varying only a small subset of the parameters showed that the technique can produce young faces that still seem natural. This was confirmed in the proportional study, which showed that it is possible to create faces with similar proportions to that of a specific person when they were a child. We also showed that the proposed parameter space can be used to rejuvenate head models of different genders and ethnic groups. The initial experiments as well as a casual examination of the algorithm have shown that the effect of the different parameters co-varies in some cases and interacts in non-linear ways in other cases. Thus, future work will focus on determining the effect of changes in the cheek area and will also evaluate the individual parameters independently by e.g. showing the participants the modification of the parameters k_i, n_i, m_i separately per facial area and let them decide which parameter value for the head shape, nose bridge, nose tip or cheeks resemble best a child at a certain age group. Further future experimental design could also allow the participants to interactively modify the parameter values for k_i, n_i and m_i for each area (e.g. in form of sliders) to make the modified 3D model match their impression of a child at a certain age. We will also focus on re-formulating the equations to allow a more intuitive selection of a desired age. Furthermore, future work should also incorporate an automatic detection of the location of the core facial regions in the initial 3D-scan to avoid a manual choice of angles. Additionally, an age-regression of the texture should be performed to fully enable a transformation into a child. Finally, since the technique captures the difference between two age groups well, it would be interesting to combine this parameter space with machine learning techniques such Scherbaum et al.'s technique [Sch07]. This would allow us to automatically learn the proper parameter values needed to regress a 3D scan of an adult into a 3D scan of a child.

7 REFERENCES

[Alb11] M. Albert, A. Sethuram and K. Ricanek. Implications of adult facial aging on biometrics. INTECH Open Access Publisher, 2011.

[Cah90] J. Cahn. The Generation of Affect in Synthesized Speech. Journal of the American Voice I/O Society., 8, 1–19, 1990.

[Enl89] D. Enlow. Handbuch des Gesichtswachstums. Quintessenz Bibliothek, 1989.

[Far94] L. Farkas Anthropometry of the head and face Raven Press., 1994.

[Fis16] K. Fisher, J. Towler and M. Eimer. Facial identity and facial expression are initially integrated at visual perceptual stages of face processing. Neuropsychologia., 80, 115–125, 2016.

[Fu10] Y. Fuo, G. Guo and T. Huang. Age Synthesis and Estimation via Faces. IEEE Transactions on Pattern Analysis and Machine Intelligence., 32, 1955 – 1976, 2010.

[Gle75] J. Gleason. Fathers and other strangers: Men's speech to young children. Developmental psy-

ISBN 978-80-86943-41-1

cholinguistics: Theory and applications., 1, 289–297, 1975.

[Gol06] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz and T. Funkhouser. A Statistical Model for Synthesis of Detailed Facial Geometry. ACM Trans. Graph. , 25(3), 1025–1034, 2006.

[Kop05] S. Kopp and L. Gesellensetter and N. Krämer and I. Wachsmuth A conversational agent as museum guide–design and evaluation of a real-world application Springer., 2005.

[Kro17] F. Kron and M. Fetter and M. Scerbo and C. White and D. Becker and others Using a computer simulation for teaching communication skills: A blinded multisite mixed methods randomized controlled trial Elsevier., 2017.

[Kru12] E. Krumhuber, M. Hall, J. Hodgson and A. Kappas. Designing interface agents: Beyond realism, resolution, and the uncanny valley. Proceedings of the 6th Workshop on Emotion and Computing–Current Research and Future Impact., 18 – 25,2012.

[Mar83] L. Mark and J. Todd. The perception of growth in three dimensions. Springer. , 33(2), 193–196, 1983.

[McD12] R. McDonnell, M. Breidt and Heinrich H. Bülthoff. Render Me Real?: Investigating the Effect of Render Style on the Perception of Animated Virtual Humans. ACM Trans. Graph., 31(4), 91:1–91:11, 2012.

[Mor15] L. Morency and G. Stratou and D. DeVault and A. Herholt and M. Lhommet and G. Lucas and F. Morbini and K. Georgila and S. Scherer and J. Gratch SimSensei Demonstration: A Perceptive Virtual Human Interviewer for Healthcare Applications. Proceedings of AAAI., 2015

[Nie09] R. Niewiadomski and E. Bevacqua and M. Mancini and C. Pelachaud Greta: An Interactive Expressive ECA System International Foundation for Autonomous Agents and Multiagent Systems., 2009.

[Par08] F. Parke and K. Waters. Computer facial animation. A.K. Peters Ltd., 2008.

[Pit75] J. Pittenger and R. Shaw. Aging faces as viscal-elastic events: implications for a theory of nonrigid shape perception. Journal of Experimental Psychology: Human perception and performance. , 1(4), 374, 1975.

[Ram06] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. IEEE Computer Vision and Pattern Recognition. , 1, 387–394, 2006.

[Ram09] N. Ramanathan, R. Chellappa and S.Biswas. Age progression in human faces: A survey. Journal of Visual Languages and Computing., 15, 3349–3361, 2009.

[Rya86] E. Ryan, H. Giles, G. Bartolucci and K. Henwood. Psycholinguistic and social psychological components of communication by and with the elderly. Language & Communication., 6(1-2), 1–24, 1986.

[Sch07] K. Scherbaum, M. Sunkel, H.-P- Seidel and V. Blanz. Prediction of Individual Non-Linear Aging Trajectories of Faces. Computer Graphics Forum. , 26(3), 285–294, 2007.

[Sha04] Z. Liu, Z. Zhang and Y. Shan. Image-based surface detail transfer. IEEE Computer Graphics and Applications., 24(3), 30–35, 2004.

[Suo07] J. Suo, F. Min, Z. Songchun, S. Shan and X. Chen. A multi-resolution dynamic model for face aging simulation. IEEE Computer Vision and Pattern Recognition, 2007. CVPR'07. , 1–8, 2007.

[Ten24] http://www.3dscanstore.com/

[Tho17] D.A.W. Thompson. On Growth and Form. Cambridge University Press. , 1917.

[Tod80] J. Todd, L. Mark, R. Shaw and J. Pittenger. The perception of human growth. Scientific American., 242, 132 – 144, 1980.

[Vin06] V. Vinayagamoorthy, M. Gillies, A. Steed, E. Tanguy, X. Pan, C. Loscos and M. Slater. Building expression into virtual characters. Proc. Eurographics Conf. State of the Art Report., 2006.

[Wu94] W. Yin, NM. Thalmann and D. Thalmann. A plastic-visco-elastic model for wrinkles in facial animation and skin aging. Proceedings of the Second Pacific Conference on Computer Graphics and Applications, Pacific Graphics '94. Fundamentals of Computer Graphics., 24(3), 201–214, 1994.

[Yar09] S. Yarosh and G. Abowd. Embodied Interaction for Mediated Communication between Children and Parents. , 2009.

 ISBN 978-80-86943-41-1

ISSN 2464-4617 (print)
ISSN 2464-4625 (CD)
Computer Science Research Notes
CSRN 2802
Short Papers Proceedings
http://www.WSCG.eu

Empirical study on label smoothing in neural networks

Mauro Mezzini
Roma Tre University
Via Castro Pretorio, 20
00182, Rome, Italy
mauro.mezzini@uniroma3.it

ABSTRACT

Neural networks are now day routinely employed in the classification of sets of objects, which consists in predicting the class label of an object. The softmax function is a popular choice of the output function in neural networks. It is a probability distribution of the class labels and the label with maximum probability represents the prediction of the neural network, given the object being classified. The softmax function is also used to compute the loss function, which evaluates the error made by the network in the classification task. In this paper we consider a simple modification to the loss function, called label smoothing. We experimented this modification by training a neural network using 12 data sets, all containing a total of about 1.5×10^6 images. We show that this modification allow a neural network to achieve a better accuracy in the classification task.

Keywords

neural networks; label smoothing; regularization; softmax; visual domain decathlon challenge;

1 INTRODUCTION

One of the most important and studied problem in artificial intelligence and computer vision is the *object classification problem* [1]. In this problem we have a set of objects, which can be images, speech, sounds and so on and we may suppose that the numerical representation of an object is a n-dimensional vector $x \in \mathbb{R}^n$. There exists a function $f : \mathbb{R}^n \to \{1,2,\ldots,K\}$, that associate to each object x a *class* $f(x)$, where $K \in \mathbb{N}$, is the number of different classes (e.g. x is an image and $f(x)$ is the subject of the image). The solution to the classification problem consists in determining a function equivalent to f. Neural networks (NN) recently achieved a very high accuracy in the classification tasks of images [2, 3, 4, 5, 6, 7]. The classification problem can also be related to other tasks such as object detection [8], image segmentation [9].

The output of the NN, is a highly complex non-linear function $z(x) \in \mathbb{R}^K$, which, in turn, is dependent on the parameters of the NN. This function is used to obtain a probability distribution of the class j given the object x, called the *softmax function*. If $z_j(x)$ is the j-th component of $z(x)$, $j = 1,\ldots,K$ then the *softmax* function is given by

$$p(j|x) = \frac{e^{z_j(x)}}{\sum_{i=1}^{K} e^{z_i(x)}} \tag{1}$$

The goal is then to obtain a softmax function such that $f(x) = \text{argmax}_j p(j|x)$. In order to do this a NN is trained using a *training set* D. Each element of D is a couple $(x_i, f(x_i))$, $x_i \in \mathbb{R}^n$, $i = 1,2,\ldots,N$ and N is the size of the training set. In order to determine the accuracy of the network, a *loss function* is employed, that assigns to each object x a quantitative measure of the error the NN made in classifying x. Often, the loss function takes the negative logarithm of the softmax function as follows

$$L(x) = -\log p(f(x)|x) \tag{2}$$

and, in order to improve the NN, the gradients (with respect to the parameters of the NN, hidden in the definition of z) of the mean of the loss function over all training set

$$L = -1/N \sum_{i=1}^{N} \log p(f(x_i)|x_i) \tag{3}$$

are back propagated along all the layers of the NN. The negative logarithm of the softmax function can be interpreted as the cross-entropy between the probability distribution given in 1 and the true probability $q(j|x)$ of the class j given the object x.

$$\mathbb{H}(q,p) = -\sum_{i=1}^{K} q(i|x) \log p(i|x) \tag{4}$$

In most of the literature regarding the cross-entropy loss function in NN, the probability distribution q is taken as follows

$$q(j|x) = \begin{cases} 1 & \text{if } f(x) = j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and having chose q as in (5), and substituting it in (4), one obtains (2) which is commonly referred to as the *categorical cross-entropy*. More in general the loss function can be interpreted as the Kullback-Leibler divergence [10] between the true probability q and p, denoted $D(q||p)$, that is

$$D(q||p) = \sum_{i=1}^{K} q(i|x) \log \frac{q(i|x)}{p(i|x)} \quad (6)$$

Note that the function given in (6) will be equivalent to (4) when we choose q as in (5).

The reason that the probability q is taken as in (5) is motivated by the obvious fact that the training set D is prepared before the actual training and who prepares the training set knows "for sure" (a priori) that the object x has a class $f(x)$. However, even who prepares the training set is subject of error and there is a degree of uncertainty for some images. For example, the CIFAR-10 [11] training set is composed by 50000 images of resolution $32 \times 32 \times 3$ and there are 10 different classes. Some images appears as a confuse blob of green and brown color. A person that looks at one of these images barely recognizes it as a frog and the classification of the image is based more on excluding that the image could not be an airplane, a car and no one of the other classes, rather than based on a certainty that the image is a frog.

One of the main problem in training a neural network is the over-fitting. The over-fitting of a neural network, is the problem that prevent the network to generalize and obtain accurate prediction on samples not contained in the training set.

Moreover and worse, one finds that, sometimes, the training process spends a lot of time to reduce the loss (3) without even achieve better fitting on the training set. With a close inspection in fact one can observe that a considerable amount of the time is spent, by the training process, on to make better an already good and accurate prediction. Ad example, suppose there are $K = 10$ different classes and take the softmax function computed by a network for an object α as follows: $p(f(\alpha)|\alpha) = 0.19$ and $p(i|\alpha) = 0.09$ for $i \neq f(\alpha)$. With this value the network accurately predicts the class $f(\alpha)$ and the value of the loss function ((3)) for the sample α is $-\log p(f(\alpha)|\alpha) = 1.6607$. However if we assign to the softmax function a different value, for example $p(f(\alpha)|\alpha) = 0.91$ we obtain a loss equal to 0.0943 which is considered better to the training process with respect to the previous value. But this new value does not improve neither the accuracy of the network on the training set nor the accuracy of the network on the test set.

So based on this observation we propose a simple modification to the cross-entropy called label smoothing. The rest of the paper is organized as follows. In Section 2 we present and briefly discuss the related works. In Section 3 we present the modification to the loss function. In Section 4 we present the results of the experiments. In Section 5 we conclude the paper.

2 RELATED WORKS

In literature numerous strategies are used to prevent the over-fitting. Data augmentation is one of the best practices employed [12, 13, 14]. In this case, each sample of the training set is modified. For example an image is manipulated by shifting it, rotating it or by changing the level of its brightness. The manipulated images are added and used in the training set. Dropout technique [15] also has been proved effective in reducing the over-fitting. In this case a random sample of neurons of a layer of the network are dropped out and the forward and backward propagation is made only on the thinned network. Batch normalization [16] is found to be a form of regularization of the network.

In [17] it has been proposed to disturb the loss layer by randomly changing the label of each sample according to a multinulli distribution. In this way the label of a sample can be different from the true label. The results of the experiments made on five different data sets show that the method effectively prevent the network in over-fitting. However the authors used in their experiment a LeNet-like [18] or AlexNet [2] models which have a shallow architecture (only 5 layers deep) and they are somewhat "older" models. In our experiments we make use of the recent ResNet model [3, 19] and a much deeper architecture on 12 different data sets.

In [20] label smoothing methods are proposed that modified the loss function by using its own prediction distribution.

The concept of label smoothing regularizations (LSR) has been investigated in [21]. They established the ground-truth probability distribution as

$$q(j|x) = (1-\varepsilon)\delta_{j,f(x)} + \varepsilon u(j)$$

where $\delta_{j,f(x)} = 1$ if $j = f(x)$ and $\delta_{j,f(x)} = 0$ otherwise, and $u(j)$ is a fixed distribution. By setting $\varepsilon = 0.1$ and $u(j) = 1/K$ they reported a gain of 0.2% in the accuracy of the Inception model on the ImageNet dataset [22].

In [23] it is proposed a regularization technique which consist in the following. If the network is over-fitting this means that the entropy of the softmax, given by

$$\mathbb{H}(p) = -\sum_{i=1}^{K} p(i|x) \log p(i|x))$$

 ISBN 978-80-86943-41-1

is low. Therefore the idea is to penalize the loss function by adding a negative entropy to the loss function as follows

$$L'(x) = -\log p(f(x)|x) - \beta \mathbb{H}(p);$$

where the parameter β control the strength of the penalization.

All the above mentioned works experimented the proposed methods either on one or two benchmark data sets or are carried on using somewhat "older" NN models. the contribution of this paper is to extend the experimentation on wide range of data sets and using the latest state-of-the-art models.

3 THE LABEL SMOOTHING

The softmax function assign to each object x and to each possible class c a value $0 \leq p(c|x) \leq 1$. In predicting the class of the object x, using the NN, we simply take $\hat{c} = \text{argmax}_c p(c|x)$; we do not care which is the value of $p(\hat{c}|x)$, we care only that $\hat{c}$ is the class such that $p(\hat{c}|x) > p(c|x)$ for all $c \neq \hat{c}$; furthermore, in measuring the accuracy of the prediction we do not care how high is the $p(\hat{c}|x)$. For example, suppose there are 10 different classes, that is $K = 10$, and suppose that $p(\hat{c}|x) = 0.2$ and $p(c|x) = 0.8/(K-1)$ for $c \neq \hat{c}$. If $\hat{c} = f(x)$ then we consider the network very accurate in predicting the class of x, even tough 0.2 is very far from 1.0. This is sometimes similar in what a human do in recognize an image. Sometimes we have not the certainty that the class of an image is a deer but we can exclude that it is a dog and it is a horse, so we classify it as a deer.

So, in order to apply this idea we use, as a loss function, the cross-entropy between the softmax function (1) and a probability distribution other than (5). If $0 \leq \gamma \leq 1$ then we may choose q as

$$q(j|x) = \begin{cases} \gamma & \text{if } f(x) = j \\ (1-\gamma)/(K-1) & \text{otherwise} \end{cases} \qquad (7)$$

with the constraint that $\gamma > 1/K$. The above methods can be implemented in existing software almost effortless. We made extensive experiments using state of the art NN for classification on several data sets. We report the results of the experiments in the following section. We found that, in many settings, there is a value of γ that improve the accuracy of the net with respect to the categorical cross-entropy.

4 THE EXPERIMENTS

We used the ten datasets of the Visual Domain Decathlon challenge presented in [24], the MNIST dataset of handwritten digit recognition [18] and the CIFAR-10 dataset. The detailed description of all these datasets is provided in [25], see also [11, 26, 27, 28, 29, 30, 31, 32]. We used Tensorflow framework [33] with Keras high-end library [34] on two Tesla P100-SXM2 GPUs. We trained each dataset with the model ResNet v2 [19] or with the model ResNet v1 [3]. In order to optimize the limited resource of GPU time and memory we choose to implement the ResNet v2 model with 83 layers and the ResNet v1 model with 20 layers. We used the last model only for the dataset ImageNet and MNIST. The optimizer method utilized for training the network is Adam [35], with initial learning rate of 0.001 which is reduced to of a factor 10^{-1} after 80, 120 and 160 epochs and of a factor of 0.5×10^{-3} after 180 epochs, for a total of 200 epochs. Each data set, with the exception of ImageNet data set, has been normalized by subtracting the mean over all the training sample.

Due to limited time [1], the training of ImageNet data set has been stopped after 100 epochs. Furthermore the model ResNeXt having 83 levels it has been trained only for $\gamma = 1$ ad with $\gamma = 0.9$. Other experiments on ImageNet were done by training a 20 layer ResNet v1 for the value of $\gamma \in \{1.0, 0.9, 0.8, 0.5\}$. In addition the minibach size for the imagenet training has been put at 64 training sample. To each image of the data set has been applied a simple data augmentation manipulation consisting in randomly shift the image, horizontally and vertically, up to 10% of the original width and height respectively, followed by a random horizontal flip.

Since the goal of the study is to asses the differences of accuracy of the network for different values of the parameter γ, we did not intend to compare our results to the best state-of-the-art models.

For each data set we tested the value of γ from 1 to a value greater than $1/K + 0.1$ where K is the number of different classes of the data set.

In Table 1 and 2 we report the results of the experiments made on the ten data sets of the Visual Domain Decathlon challenge, the MNIST and CIFAR-10 data sets. The value of γ ranged from 1.0 to $1/K + 0.1$ with step value of 0.1. For the value of γ and for each data set, is reported the best accuracy attained on the validation set. We can see that there are sometimes dramatic improvement in the accuracy in some of the data set of the experiment as reported in Table 3 in which it is compared the value of the accuracy with $\gamma = 1.0$ and the best accuracy obtained among all different values of γ. We can observe that one limitation of our approach is that the value of γ, for which the best accuracy is attained, is not the same on the various data sets making difficult to adapt this method to other cases.

[1] On two Tesla P100-SXM2, each epoch require more than 8000 seconds to terminate

 ISBN 978-80-86943-41-1

	γ				
Dataset	**0.10**	**0.20**	**0.30**	**0.40**	**0.50**
aircraft	24.82	36.43	39.08	40.82	41.03
cifar100	59.25	66.76	69.06	71.20	71.39
daimlerp.	-	-	-	-	-
dtd	28.21	27.78	25.92	26.98	25.49
gtsrb	99.68	99.78	99.81	99.86	99.83
omniglot	82.87	83.50	84.22	84.75	85.21
svhn	-	92.41	93.05	93.64	93.62
ucf101	64.79	72.78	74.37	75.91	74.88
vgg-flow.	51.03	52.80	50.93	49.66	48.09
ImageN.(a)	-	-	-	-	-
ImageN.(b)	-	-	-	-	35.59
cifar10	-	86.98	88.77	90.33	90.76
mnist	-	99.41	99.46	99.61	99.54

Table 1: For the value of $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and for each dataset is reported the best accuracy attained on the validation set. The data sets daimlerpedcls, $K = 2$ classes while svhn, cifar10, and mnist have $K = 10$ classes. Therefore there are no data for the value of $\gamma \leq 1/K$ on these data sets. The model (a) used for ImageNet is ResNet v2 with 83 layers trained for 100 epochs, while the model (b) is ResNet v1 with 20 layers trained for 200 epoch. The model used for mnist is ResNet v1 with 20 layers.

	γ				
Data set	**0.60**	**0.70**	**0.80**	**0.90**	**1.00**
aircraft	38.06	41.09	36.10	38.63	34.75
cifar100	71.80	71.88	71.73	71.34	70.83
daimlerp.	99.80	99.83	99.97	99.91	99.98
dtd	25.39	25.23	23.42	23.68	25.28
gtsrb	99.86	99.85	99.82	99.87	99.94
omniglot	85.21	84.61	83.28	82.01	77.95
svhn	93.79	93.95	94.07	94.22	93.82
ucf101	74.37	74.17	74.17	72.17	70.12
vgg-flow.	46.32	48.48	45.83	44.26	42.30
ImageN.(a)	-	-	-	45.93	42.87
ImageN.(b)	-	-	43.00	43.95	44.76
cifar10	91.60	91.95	92.20	92.54	93.31
mnist	99.53	99.55	99.55	99.61	99.56

Table 2: For the value of $\gamma \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$ and for each data set is reported the best accuracy attained on the validation set. The model (a) used for ImageNet is ResNet v2 with 83 layers trained for 100 epochs, while the model (b) is ResNet v1 with 20 layers trained for 200 epoch. The model used for mnist is ResNet v1 with 20 layers.

Data set	**Best**	**Acc.** $\gamma = 1$	**diff** %
aircraft	41.09	34.75	15.41
cifar100	71.88	70.83	1.46
daimlerp.	99.98	99.98	0.00
dtd	28.21	25.28	10.38
gtsrb	99.94	99.94	0.00
omniglot	85.21	77.95	8.52
svhn	94.22	93.82	0.42
ucf101	75.91	70.12	7.63
vgg-flow.	52.80	42.30	19.89
ImageN.(a)	45.93	42.87	7.12
ImageN.(b)	44.76	44.76	0.00
cifar10	93.31	93.31	0.00
mnist	99.61	99.56	0.05

Table 3: The best accuracy achieved compared to the accurcay of the model trained with $\gamma = 1$.

5 CONCLUSIONS

We proposed a simple and easy to implement method of regularization of the model, called label smoothing, and we made extensive experiments on several data sets using state-of-the art very deep models. We showed that this method may be very effective in regularize the model and mitigate over-fitting. Future researches may investigate to mix the method proposed in this paper with the ones proposed in [17, 23], and at the same time, extending the experiments to different models.

6 REFERENCES

[1] Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: S. P. Singh, S. Markovitch (Eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., AAAI Press, 2017, pp. 4278–4284.

[5] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision

(ICCV), 2015, pp. 1026–1034. doi:10.1109/ICCV.2015.123.

[6] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 00, 2017, pp. 6450–6458. doi:10.1109/CVPR.2017.683.

[7] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, CoRR abs/1709.01507.

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[9] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[10] S. Kullback, R. A. Leibler, On information and sufficiency, Ann. Math. Statist. (1) 79–86. doi:10.1214/aoms/1177729694.

[11] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep. (2009).

[12] J. Salamon, J. P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, IEEE Signal Processing Letters 24 (3) (2017) 279–283. doi:10.1109/LSP.2017.2657381.

[13] P. Y. Simard, D. Steinkraus, J. Platt, Best practices for convolutional neural networks applied to visual document analysis, Institute of Electrical and Electronics Engineers, Inc., 2003.

[14] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, CoRR abs/1712.04621.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[16] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: ICML, 2015.

[17] L. Xie, J. Wang, Z. Wei, M. Wang, Q. Tian, DisturbLabel: Regularizing CNN on the loss layer, Vol. 2016-December, IEEE Computer Society, 2016, pp. 4753–4762. doi:10.1109/CVPR.2016.514.

[18] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, D. Henderson, Advances in neural information processing systems 2, 1990, Ch. Handwritten Digit Recognition with a Back-propagation Network, pp. 396–404.

[19] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, 2016, pp. 630–645.

[20] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, A. Rabinovich, Training deep neural networks on noisy labels with bootstrapping, CoRR abs/1412.6596.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, CoRR abs/1512.00567.

[22] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[23] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, G. E. Hinton, Regularizing neural networks by penalizing confident output distributions, CoRR abs/1701.06548.

[24] 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017.

[25] Visual domain decathlon, 2017.
URL http://www.robots.ox.ac.uk/~vgg/decathlon/

[26] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, Tech. rep. (2013). arXiv:1306.5151.

[27] S. Munder, D. M. Gavrila, An experimental study on pedestrian classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006) 1863–1868. doi:10.1109/TPAMI.2006.217.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, 2011.

[29] M. E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing, 2008, pp. 722–729. doi:10.1109/ICVGIP.2008.47.

[30] Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, Neural Networks 32 (2012) 323 – 332, selected Papers from IJCNN 2011. doi:https://doi.org/10.1016/j.neunet.2012.02.016.

 ISBN 978-80-86943-41-1

[31] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction 350 (6266) (2015) 1332–1338. doi:10.1126/science.aab3050.

[32] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, CoRR abs/1212.0402.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A system for large-scale machine learning, 2016.

[34] F. Chollet, et al., Keras, https://github.com/keras-team/keras (2015).

[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization., CoRR.