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

CSRN 2502

(Ed.)

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

Computer Science Research Notes

23rd International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2015
Plzen, Czech Republic
June 8 - 12, 2015

Proceedings

WSCG 2015

Short Papers Proceedings

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

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

CSRN 2502

(Ed.)

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

Computer Science Research Notes

23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2015
Plzen, Czech Republic
June 8 - 12, 2015

Proceedings

WSCG 2015

Short Papers Proceedings

Vaclav Skala – Union Agency

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

Computer Science Research Notes
CSRN 2502

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

Managing Editor: Vaclav Skala

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

ISSN 2464-4617 (Print) ***ISSN 2464-4625*** (CD/DVD)
ISBN 978-80-86943-66-4 *(Print)* ***ISBN 978-80-86943-62-6*** *(*CD/-ROM)

WSCG 2015

International Program Committee

Andrew, Glassner (United States)
Baranoski, Gladimir (Canada)
Benes, Bedrich (United States)
Benger, Werner (Austria)
Bengtsson, Ewert (Sweden)
Bourke, Paul (Australia)
Dachsbacher, Carsten (Germany)
Daniel, Marc (France)
Daniels, Karen (United States)
Debelov, Victor (Russia)
Feito, Francisco (Spain)
Ferguson, Stuart (United Kingdom)
Gavrilova, Marina (Canada)
Guthe, Michael (Germany)
Jung, Soon Ki (Korea)
Kalra, Prem K. (India)
Klosowski, James (United States)
Kraus, Martin (Denmark)
Linsen, Lars (Germany)
Lu, Aidong (United States)
Mark, Finch (United States)
Molla, Ramon (Spain)
Muller, Heinrich (Germany)
Murtagh, Fionn (United Kingdom)
Oyarzun Laura, Cristina (Germany)
Pan, Rongjiang (China)
Paquette, Eric (Canada)
Patow, Gustavo (Spain)
Pedrini, Helio (Brazil)
Platis, Nikos (Greece)
Renaud, Christophe (France)
Richardson, John (United States)
Rojas-Sola, Jose Ignacio (Spain)
Ruyam, Acar (Turkey)
Segura, Rafael (Spain)
Semwal, Sudhanshu (United States)
Schultz, Thomas (Germany)
Schulz, Hans-Jorg (Germany)
Sousa, A.Augusto (Portugal)
Stroud, Ian (Switzerland)
Szecsi, Laszlo (Hungary)
Teschner, Matthias (Germany)
Tevfik, Akgun (Turkey)
Tokuta, Alade (United States)
Ugur, Gudukbay (Turkey)
Wu, Shin-Ting (Brazil)
Wuensche, Burkhard,C. (New Zealand)
Wuethrich, Charles (Germany)
Zemcik, Pavel (Czech Republic)
Zwettler, Gerald (Austria)

Board of Reviewers

Meng, Weiliang (China)
Menotti, David (Brazil)
Mestre, Daniel,R. (France)
Meyer, Alexandre (France)
Michael, Despina (Cyprus)
Michels, Dominik (United States)
Monti, Marina (Italy)
Montrucchio, Bartolomeo (Italy)
Movania, Muhammad Mobeen (Pakistan)
Mukai, Tomohiko (Japan)
Mura, Claudio (Switzerland)
Nagai, Yukie (Japan)
Nah, Jae-Ho (Korea)
Nanni, Loris (Italy)
Nogueira, Keiller (Brazil)
Nurzynska, Karolina (Poland)
Nyul, Laszlo (Hungary)
Oliveira, Joao Fradinho (Portugal)
Oztimur Karadag, Ozge (Turkey)
Paiva, Jose Gustavo (Brazil)
Parsons, Paul (Canada)
Patane, Giuseppe (Italy)
Paul, Padma Polash (Canada)
Peethambaran, Jiju (India)
Penedo, Manuel (Spain)
Pina, Jose Luis (Spain)
Pobegailo, Alexander (Belarus)
Puig, Anna (Spain)
Ramos, Sebastian (Germany)
Rasool, Shahzad (Singapore)
Reddy, Pradyumna (India)
Rehfeld, Stephan (Germany)
Rind, Alexander (Austria)
Rupprecht, Christian (Germany)
Sadlo, Filip (Germany)
Saito, Shunsuke (United States)
Santagati, Cettina (Italy)
Saraiji, MHD Yamen (Japan)
Saru, Dhir (India)
Seipel, Stefan (Sweden)
Shesh, Amit (United States)
Shi, Xin (China)
Shimshoni, Ilan (Israel)
Schaefer, Gerald (United Kingdom)
Schmidt, Johanna (Austria)
Schultz, Thomas (Germany)
Schwarz, Michael (Switzerland)
Silva, Romuere (Brazil)
Silva, Samuel (Portugal)
Singh, Rajiv (India)
Solis, Ana Luisa (Mexico)
Soriano, Aurea (Brazil)
Souza e Silva, Lucas (Brazil)
Spiclin, Ziga (Slovenia)
Svoboda, Tomas (Czech Republic)
Tavares, Joao Manuel (Portugal)
Teixeira, Raoni (Brazil)
Theussl, Thomas (Saudi Arabia)
Tomas Sanahuja, Josep Maria (Mexico)
Torrens, Francisco (Spain)
Tytkowski, Krzysztof (Poland)
Umlauf, Georg (Germany)
Vasseur, Pascal (France)
Vazquez, David (Spain)
Veras, Rodrigo (Brazil)
Walczak, Krzysztof (Poland)
Wanat, Robert (United Kingdom)
Wang, Lili (China)
Wang, Ruizhe (United States)
Wang, Lisheng (China)
Wenger, Rephael (United States)
Wijewickrema, Sudanthi (Australia)
Wu, YuTing (Taiwan)
Wu, Jieting (United States)
Wuensche, Burkhard,C. (New Zealand)
Xiong, Ying (United States)
Xu, Tianchen (Hong Kong SAR)
Xu, Chang (China)
Yang, Shuang (China)
Yasmin, Shamima (United States)
Yoshizawa, Shin (Japan)
Yu, Hongfeng (United States)
Zheng, Jianping (United States)
Zhong, Li (China)

WSCG 2015

Short Papers Proceedings

Contents

New algorithms for satellite data verification with and without the use of the imaged area vector data

Andrey Kuznetsov
Samara State Aerospace University
Moskovskoye shosse, 34
Russia, Samara
kuznetsoff.andrey@gmail.com

Vladislav Myasnikov
Samara State Aerospace University
Moskovskoye shosse, 34
Russia, Samara
vmyas@geosamara.ru

ABSTRACT

This paper presents a solution of remote sensing data verification problem. Remote sensing data includes digital image data and metadata, which contain parameters of satellite imaging process (Sun and satellite azimuth and elevation angles, creation time, etc.). The solution is based on the analysis of special numerical characteristics, which directly depend on the observation parameters: sun position, satellite position and orientation. These characteristics are based on model-oriented descriptor, proposed by one of the co-authors of this paper. We propose two fully automatic algorithms for remote sensing data analysis and decision-making based on data compatibility: the first one uses vector data of the imaged area as a prior information, the second doesn't. After algorithms description we provide results of conducted experiments and explain appliance limits of the proposed algorithms.

Keywords

Satellite digital image, vector map, shadow buffer zone, model-oriented descriptor, amplitude-phase mismatch, Canny edge detector, edge tracing

1. INTRODUCTION

Widely used in the modern world remote sensing data (RSD) consist of two main components: a digital image and its metadata, which describe the process and the observation parameters. During RSD transmission from source to destination, this data can be distorted accidentally (due to errors) and intentionally (by hackers). When this happens, the satellite image itself and/or its metadata can be changed. The problem of forgery detection in digital images, when observation parameters and image metadata are not used or unknown, is being solved in [Chr12a, Glu11a, Far09a, Far09b, Kuz14a].

Nowadays, there are papers devoted to the analysis of light parameters inconsistency for local parts of a single object in digital images [Mya12a]. These algorithms use only image data during analysis, because additional information about observation parameters is absent (the research is carried out for digital images obtained by ordinary cameras that do not store observation information). Due to the lack of this data, there is nothing to compare with angles and lengths of shadows in the analyzed image. Metadata of satellite images and imaged area vector maps allow to analyze the consistency of objects and their shadows. During literature analysis there were not found any papers aimed at the detection of inconsistency in shadows and objects in satellite images.

In this paper we propose a new solution for detection of digital satellite image and observation parameters inconsistency using model-oriented descriptors, proposed in papers [Mya12a, Mya12b] by one of the co-authors of this work.

2. PROBLEM DEFINITION

To identify irrelevance between an image and its metadata, we will analyze the shadows of tall objects on the image. There will be used buildings with height of at least 12 meters (for example, houses with 5 floors and more), which have a simple rectangular form on a satellite image received by nadir observation. The length of the analyzed shadows of such a building is 10-15 m - if the length exceeds this value the shadows may be imposed on neighboring buildings (in dense urban areas), which may impair analysis quality. It is better to identify objects and their shadows with such linear characteristics on high-resolution images (0.5-1 m). This is why we will use Geoeye-1 satellite images (spatial resolution – 0.5 m). This parameters characterize the restrictions wherein the performed algorithms will work correctly.

Image metadata contains the following observation parameters of the satellite image:

1. image coordinates $\mathbf{s} = (s_1, s_2, ..., s_k)^T$, where $s_i = (x_i, y_i)$ is a reference point of a satellite image, k is a number of reference points;
2. satellite position coordinates $\mathbf{p} = (\varphi_{az}, \varphi_{el}, h_{alt})$, where φ_{az} is the azimuth incidence angle of a satellite's sensor, φ_{el} is the elevation incidence angle of a satellite's angle, h_{alt} corresponds to the altitude value of a satellite;
3. Sun position coordinates $\boldsymbol{\alpha} = (\alpha_{az}, \alpha_{el})^T$, where α_{az}, α_{el} are azimuth and elevation incidence Sun angles respectively.

Fig. 1 shows the relative position of azimuth and zenith angles of Sun and spacecraft.

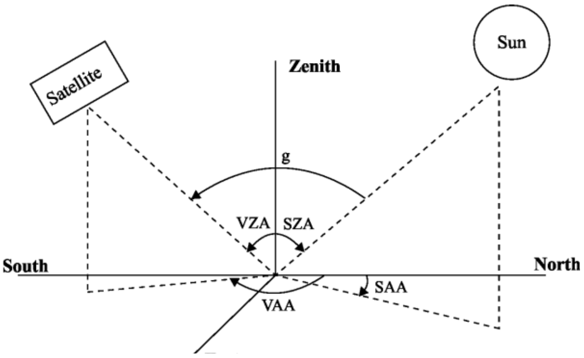


Figure 1. Arrangements of the angles for Sun and spacecraft (VAA – azimuth angle, VZA – zenith angle, SAA – azimuth Sun angle, SZA – zenith Sun angle, g – phase angle).

3. AMPLITUDE-PHASE MISMATCH

Model oriented descriptor of a digital image was proposed in [Mya12a, Mya12b]. It is a new descriptor type, which is formed on the basis of differential and probabilistic properties of the local neighborhood of the analyzed image [Mya12b].

At the heart of the model-oriented descriptor is the use of gradient field probability distribution that describes the model of the analyzed image fragment. Descriptor's components for a particular image area are calculated as the values of the probability density of a specific gradient field or its individual components. Such specificity of the proposed descriptor calculation allows to classify it as a *model-oriented* and to use it as a part of some classifier's decision rule or as a numerical characteristic of an image local area. For some image processing problems solution it is convenient not to use descriptor components, but its derivative values, called descriptor features, which were introduced in [Mya12b]. As it is shown in this work, all the proposed descriptor features have a useful property – their possible values lie in the range [0, 1]. This means that larger values correspond to greater similarity of particular image part (and, as a consequence of its gradient field) to the potentially possible realizations of the gradient field (the model). For a number of standard models of the random gradient field there were obtained explicit expressions for model-oriented descriptor features [Mya12b]. One of these models and the corresponding feature (amplitude-phase mismatch) are used later in this work [Mya12a, Mya12b].

The base of a model-oriented descriptor is the use of probability distribution of the gradient field, which characterizes the model of the analyzed image fragment. Values of descriptor's components for a particular image fragment are calculated as the values of probability density of the argument in the form of a specific gradient field or some of its components.

For a formal definition of this descriptor, we introduce some notation. Let D be an analyzed image area (area of some real object's shadow), for which the function $\varphi(t_1, t_2)$ is defined. The values of this function define orientation (angle) of a brightness difference line (along shadow's boundaries) in the corresponding position (t_1, t_2).

We will call the following equation as an amplitude-phase mismatch (APM) ζ for an image area D:

$$\zeta = \frac{SGD}{SGM}, \zeta \in [0,1], \qquad (1)$$

where *SGD* and *SGM* are represented in the form:

$$SGM = \sum_{(t_1,t_2)\in D} |g(t_1,t_2)|,$$

$$SGD = \sum_{(t_1,t_2)\in D} |g(t_1,t_2)| \left(\frac{\cos(\varphi(t_1,t_2) - \arg(g(t_1,t_2))) + 1}{2} \right).$$

At this point $g(t_1, t_2)$ is a concrete implementation of the gradient field for the given image fragment, $|g(t_1, t_2)|$ and $\arg(g(t_1, t_2))$ are its modulus and direction (phase) respectively. It is obvious that the closer APM's value ζ to 1, the more image area D matches a template, represented by $\varphi(t_1, t_2)$ function. APM, in fact, shows how far is the real gradient direction value from $\varphi(t_1, t_2)$ direction.

4. PROPOSED SOLUTION

Image and observation parameters verification procedure with imaged area vector map use

Let us consider a situation when there is a priori information about the imaged area – a vector map of this area. By carrying out a geometric calibration of a snapshot and putting it on a vector map of the imaged area, it is possible to determine positions of physical objects on the space image. Depending on the angle φ_{el} the roofs of the objects in the image can be displaced according to the spacecraft inclination angle, whereas the vector objects correspond to the foundation of these buildings and are situated as it would be for nadir satellite imaging. The example of combining the space image received from Geoeye-1 (0.5 m) satellite and the vector map of the imaged area is presented in Fig. 2.

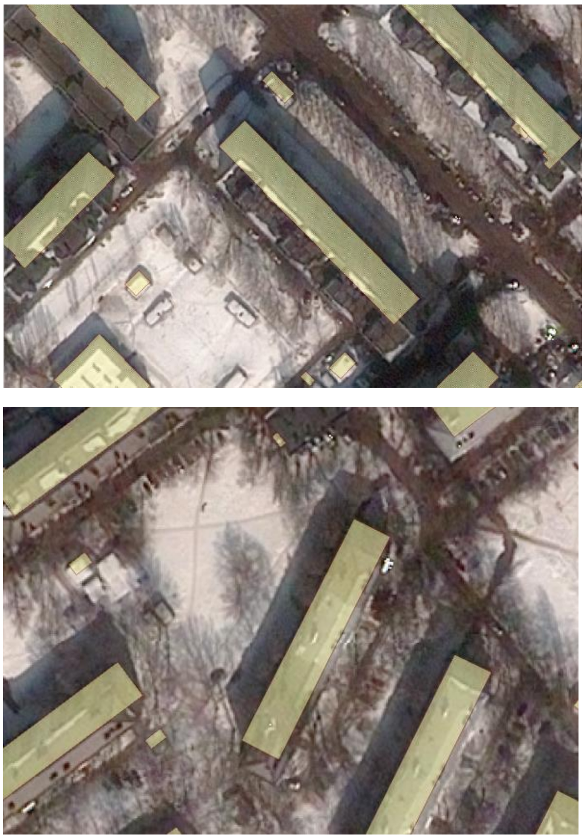


Figure 2. Satellite image and vector map combination for analyzed objects, $\varphi_{el} = 35$.

From now on we will neglect geolocation accuracy for the proposed solution description and conducted experiments.

Using semantic data of the buildings vector layer we select only those buildings, which height is more than 10 meters $h_b > 10$ (buildings height values are listed in the semantic data of the vector layer). The contour of each building is described by four points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$. The distance between any two points will be denoted as:

$$d_i^j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} .$$

Let us define the Sun position in the satellite image coordinate system (x_{sun}, y_{sun}) so, that

$$d_i^{sun} >> \max_{i,j} d_i^j, i, j \in \overline{1,4}, i \neq j .$$

For the analyzed building it is necessary to determine the pair of its sides for which the thrown shadow angle remains right (the angle whose vertex is the farthest from the Sun, e.g., B'A'C'):

$$i_{\max} = \arg\max_{i \in \overline{1,4}} d_i^{sun} .$$

There is shown an example (Fig. 3) of an object ABDC and its thrown shadow with length s (we assume that the imaging process was carried out at nadir point). The shadow is thrown by the sides AB and AC.

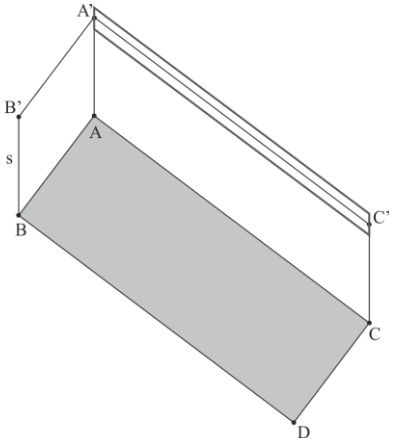


Figure 3. Shadow buffer zone for one side of the building.

Let us determine the length of the shadow of the object s as $s = h_b tg\alpha_{el}$. We then can determine the buffer zone of the shadow boundaries – geometric locus at the edge of the shadow thrown by a building side (there is allocated a buffer zone for the shadow edge A'C' in Fig. 3). In the shown in Fig. 3 case the buffer zone is a parallelogram with two sides parallel to AC, and the other two belong to a line lying at the shadow inclination angle α_s (calculated with respect to the direction of the X axis of the rectangular coordinate system of the analyzed image). We will call the shadow angle of the longer side of the parallelogram as *the direction of the buffer zone boundary*. The buffer zone for the whole building shadow lies along the polyline BB'A'C'C and consists of 4 parts: along BB', B'A', A'C' and C'C.

Let *L* be the buffer zone height (see Fig. 4), then the length of the parallelogram sides, which limit the shadow side of the buffer zone A'C ', is calculated as follows:

$$d_{A'_-}^{A'_+} = H = \frac{L}{\sin\beta},$$

where $\beta = \alpha - \alpha_S$.

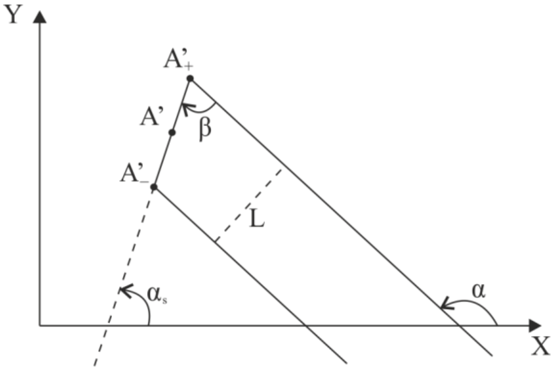


Figure 4. Shadow buffer zone edge calculation for a building side.

Shadow buffer zone borders, parallel to the side AC, will be separated by the distances $s - \frac{H}{2}$ and $s + \frac{H}{2}$ from the side AC. Coordinates of the buffer zone corners are calculated in a trivial way.

Doing similar calculations for the other three shadow borders, we get coordinates of the corners of their buffer zones. The result of the building shadow buffer zone construction (*D*) is shown in Fig. 5.

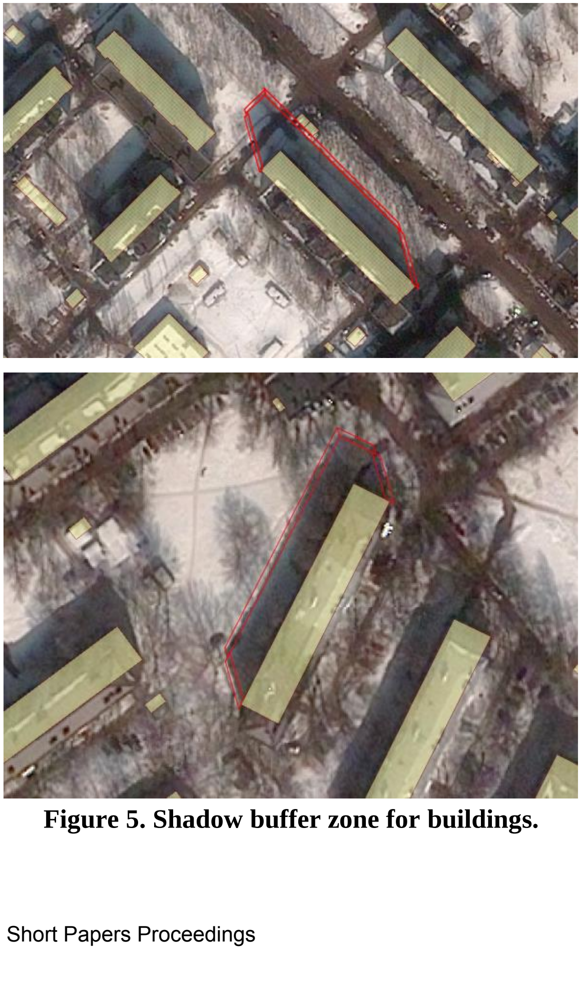


Figure 5. Shadow buffer zone for buildings.

For each of the buffer zones we will calculate APM values (1), which characterize correspondence of the real object's shadow in the satellite image (according to the orientation of the buffer zone) to the value, calculated using metadata parameters.

Image and observation parameters verification procedure without imaged area vector map use

If there is no vector map of the imaged area we need to detect buildings and corresponding shadows using only image analysis methods. We will use high resolution images for analysis as in the previous algorithm. In this paper we propose the algorithm that allows to identify the corresponding buildings corners and shadows thrown by these corners using Canny detector [Gas03a, Can86a]. This method provides precise detection results for noisy images and detected edges are one pixel in width, which enables to trace them further [Ren02a].

Let us take the following image for analysis (see Fig. 6):

$$f(m,n), m \in [0, M), n \in [0, N),$$

where *M,N* are image linear dimensions of the image.

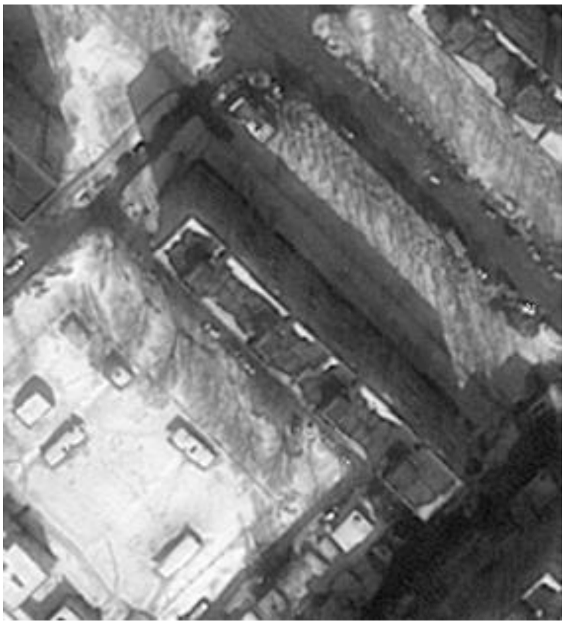


Figure 6. Analyzed image part.

Before its analysis it is necessary to make some pre-processing steps:

1) convert the image to grayscale (if it is multichannel);

2) filter noise to smooth the edges.

The image $f'(m,n)$ is a result of the above preprocessing operations.

After that we apply Canny edge detection algorithm to the preprocessed image (by means of *OpenCV*). For detector configure there are used two parameters: the first one is used to select the most significant boundaries (th_1), the second one is used to combine edge segments into contours (th_2). In this paper we use empirically selected parameters for edges detection $th_1 = 50, th_2 = 120$. These values provide the best precision for edges detection. The result of Canny edge detector will be denoted as $c_{f'}(m,n)$.

The algorithm of Canny edge detection result $c_{f'}(m,n)$ for image verification consists of two steps:

1) detection of corresponding angles of buildings roofs and shadows thrown by them;

2) detection of shadows edges parts that are collinear with shadow inclination angle, calculated using the values of analyzed image metadata.

The first step of the proposed algorithm include detection of angles between the edges that are close to 90°. The closeness of these values will be determined by a threshold parameter $\Delta_{rightAngle}$. Each building has a right angle of the roof, which corresponds to a right angle of its shadow. For each edge pixel (x_b, y_b) we produce eight-connected tracing procedure [Ren02a] in opposite directions and estimate the angle γ between these traced edge parts. If the following condition $\gamma \in \left[\frac{\pi}{2} - \Delta_{rightAngle}\ , \frac{\pi}{2} + \Delta_{rightAngle}\right]$ is satisfied, then (x_b, y_b) point is placed in the list of points, which may be a building roof angle or a shadow angle. Then the points list is filtered and only those pairs of points $(x_1, y_1),(x_2, y_2)$ are selected for which the following condition is fulfilled:

$$\left| arctg2\left(\frac{y_1 - y_2}{x_1 - x_2}\right) - \alpha_s \right| < \Delta_{sun}\ .$$

There is also taken into account the minimum and maximum possible heights of buildings, which depend on the length of objects shadows.

The result of detection of buildings roofs and thrown shadows corresponding angles is presented in Fig. 7.

The second step of the proposed algorithm is to identify edges of the shadows, which direction coincides with shadow inclination angle, calculated using the values of analyzed image metadata. In the basis of this operation also lies the tracing of $c_{f'}(m,n)$. Let K_s be a restriction on the maximum pixel length of the traced edge. When we determine a list of K_s points for a given point, we approximate it a line [Gas03a] using *Line2DFitting* function of *OpenCV*. As a result we obtain a point (x_{line}, y_{line}) belonging to this line and line direction vector (d_x, d_y). The result of this operation is presented in Fig. 8.

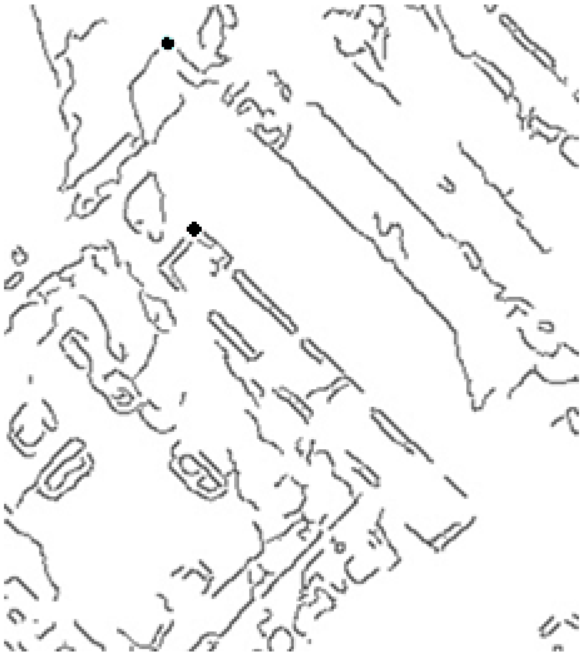


Figure 7. Corresponding angles detection for a test building.

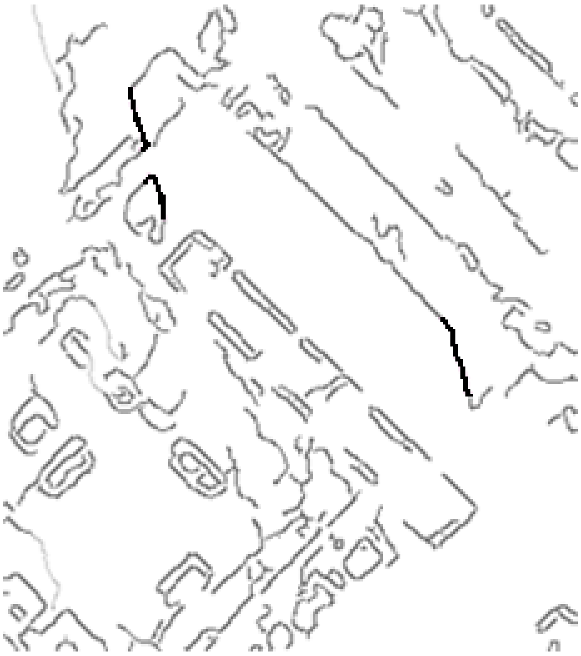


Figure 8. Detection of shadow edges collinear to metadata shadow angle.

Using the list of corresponding right angles and the list of shadow edges closest to them there is formed a geometric model of the building shadow for which the APM value is calculated, as described in the previous subsection.

5. EXPERIMENTAL RESULTS

During the algorithm research we define APM threshold values, which will be used for a decision making of satellite image to observation parameters correspondence. To conduct an experiment we choose Geoeye-1 satellite images (0.5 m resolution) and a set of 26 vector objects, randomly selected among the objects belonging to the territory of the snapshot. Taking into account the randomness of objects selection, some shadow buffer zones boundaries may appear in the shadow region of other vector objects, or may be blocked by other objects in the image.

APM values are calculated for any image channels in two ways:

1) APM value is calculated for each side of the shadow buffer zone of the object, so a training sample will consist of 104 variables – 4 values for each vector object;

2) APM value is calculated for the whole shadow buffer zone of the object - the size of a training sample will be 26 variables.

Shadow buffer zone boundaries are calculated for a given correct shadow inclination angle $\alpha_s = 75^\circ$, which was calculated based on the satellite image metadata parameters.

Let L be the volume of a training sample, then the APM threshold value for the i-th training sample method is defined as follows:

$$t_i = \min_{k \in L} \{\zeta_k\} \cdot 0.9,\ i \in \{1,2\},$$

where ζ_k is the APM value for k-th object of a training sample, 0.9 is a constant defined experimentally.

Fig. 9 and Fig. 10 show the distribution of APM values for both techniques of creating a training sample and the corresponding threshold values. Thresholds in the figures are as follows $t_1 = 0.34$ and $t_2 = 0.6$.

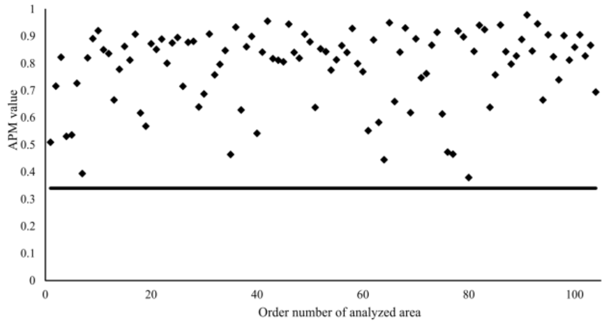


Figure 9. APM values distribution for the first creating technique.

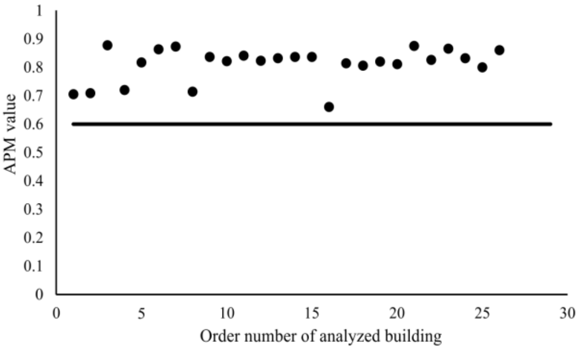


Figure 10. APM values distribution for the second creating technique.

Decision making of satellite image to observation parameters correspondence is performed as follows. There is selected a test sample of 20 buildings (vector objects) for the analyzed satellite image. On the first stage APM values are calculated for each element of the shadow buffer zone and the object doesn't pass a test if:

$$\exists\ j \in \overline{0,3},\ \ \zeta_j < t_1 \tag{2}$$

During the second stage APM values are calculated for the entire shadow buffer zone and the decision is made in a similar way:

$$\zeta < t_2 \tag{3}$$

Satellite image does not pass the validation test, if at least one test sample object does not pass a two-stage test procedure (2) – (3).

In order to confirm the correctness of APM threshold values t_1, t_2 selection we take a satellite image and a test sample of 20 vector objects, which belong to the territory of the snapshot. We then construct a relationship between the values of shadow inclination angle and the number of objects that did not pass the two-stage procedure of satellite image validation (see Fig. 11).

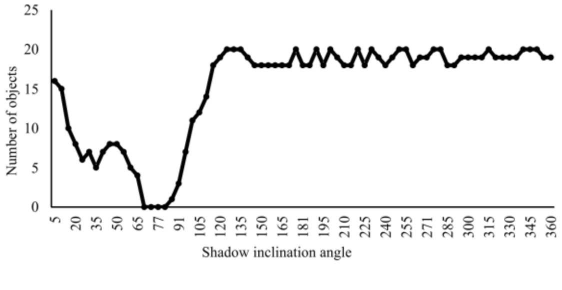


Figure 11. Dependency of test sample objects number that failed validation test from shadow inclination angle.

For presented in Fig. 6 buildings the APM value $\zeta_1 = 0.77, \zeta_2 = 0.93$ exceeds the threshold APM value. This allows to make a decision that this object corresponds to observation parameters.

According to the results of conducted experiments it can be concluded that the developed algorithm detects inconsistency of a satellite image and its observation parameters when the deviation of shadow inclination angle from its correct value is $\Delta_{\alpha_s} > 5^\circ$ for the calculated APM threshold values t_1, t_2. This is acceptable for the analysis of satellite images.

Both of the proposed algorithms have low computational complexity and can be used for real-time satellite image analysis (Geoeye-1 satellite image with size 10000×10000 pixels and 1000 vector objects average analysis time is 5400 ms on Intel Core i5 3470, 8Gb RAM).

6. CONCLUSION

In this paper we presented new algorithms for detecting inconsistencies of satellite image data and its observation parameters: with and without the use of imaged area vector map. The proposed solution makes it possible to detect inconsistencies of objects and observation parameters at a deviation angle greater than 5°. The paper also provides recommendations on parameters choice and detection algorithms usage limits. Further we are going to compare different shadow detection algorithms as one of the steps of the proposed solution and to develop an algorithm for buildings with more complex geometry.

7. ACKNOWLEDGMENTS

This work was partially supported by the Russian Foundation of Basic Research (projects: 13-07-12103-ofi, 15-07-01164-a), by the Fundamental research program № 15 of the Russian Academy of Sciences Presidium and by the Ministry of Education and Science (state order 2298).

8. REFERENCES

[Can86a] Canny, J. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI. Volume 8, Issue 6, pp. 679-698, 1986.

[Chr12a] Christlein, V., Riess, C., Jordan, J., Riess, C., and Angelopolou, E. An Evaluation of Popular Copy-Move Forgery Detection Approaches. IEEE Transactions on Information Forensics and Security. Volume 7, No. 6, pp. 1841-1854, 2012.

[Gas03a] Gashnikov, M., Glumov, N., Ilyasova, N., Myasnikov, V. [et al]. Methods of computer image processing (2-nd edition reviewed). Moscow: “Fizmatlit Publisher”. 784 p., 2003.

[Glu11a] Glumov, N., Kuznetsov, A., and Myasnikov V. The algorithm for copy-move detection on digital images. Volume 7, No.3, pp. 360-367, 2013.

[Far09a] Farid, H. Image Forgery Detection. IEEE Signal processing magazine, pp. 16-25, 2009.

[Far09b] Farid, H. Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security. Volume 1, No. 4, pp. 154-160, 2009.

[Kuz14a] Kuznetsov, A., and Myasnikov V. A fast plain copy-move detection algorithm based on structural pattern and 2D Rabin-Karp rolling hash. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 8814, Issue 1, pp. 461-468, 2014.

[Mya12a] Myasnikov, V. Method for detection of vehicles in digital aerial and space remote sensed images. Computer optics. Volume 36, No. 3, pp. 429-438, 2012.

[Mya12b] Myasnikov, V. Model-based gradient field descriptor as a convenient tool for image recognition and analysis. Computer optics. Volume 36, No. 4, pp. 596-604, 2012.

[Ren02a] Ren, M., Yang, J., and Sun, H. Tracing boundary contours in a binary image. Image and Vision Computing. Volume 20, Issue 2, pp. 125-131, 2002.

A framework for robust object multi-detection with a vote aggregation and a cascade filtering

Grzegorz Kurzejamski
Institute of Microelectronics and Optoelectronics
Warsaw University of Technology
00-661 Warsaw, Poland
Lingaro Sp. z o.o.
Puławska 99a
02-595 Warsaw, Poland
grzegorz.kurzejamski@gmail.com

Jacek Zawistowski
Institute of Microelectronics and Optoelectronics
Warsaw University of Technology
00-661 Warsaw, Poland
Lingaro Sp. z o.o.
Puławska 99a
02-595 Warsaw, Poland
jzawisto@gmail.com

Grzegorz Sarwas
Lingaro Sp. z o.o.
Puławska 99a
02-595 Warsaw, Poland
grzegorz.sarwas@gmail.com

ABSTRACT

This paper presents a framework designed for the multi-object detection purposes and adjusted for the application of product search on the market shelves. The framework uses a single feedback loop and a pattern resizing mechanism to demonstrate the top effectiveness of the state-of-the-art local features. A high detection rate with a low false detection chance can be achieved with use of only one pattern per object and no manual parameters adjustments. The method incorporates well known local features and a basic matching process to create a reliable voting space. Further steps comprise of metric transformations, graphical vote space representation, two-phase vote aggregation process and a cascade of verifying filters.

Keywords

Computer Vision, Image Analysis, Multiple Object Detection, Object Localization, Pattern Matching.

1 INTRODUCTION

As computer vision algorithms are being vastly developed in many fields, it is still very unlikely to create production-class detection systems for various applications. This paper is focused on the problem of detection of retail products shown on the market shelves and displays. This particular application demands usage of a multi-object multi-detection system (possible many instances of the different object classes in one scene). The patterns in this case are generic graphics most of the time and geometric transformations in the scene are much simpler than those found in natural scenery. Even though, it's still a demanding task as brands' numbers are counted in hundreds and each brand can have up to a thousand different wrapping layouts. Moreover, each brand has some percentage of common graphics present, for example logos. There are no standards in size or shape of the products. It's very expensive to take dozens of photos of each sample wrapping in different environmental conditions as well, so learning methods may be inefficient in real applications.

There are many approaches to multi-detection systems with generic graphics as patterns. The most common is the local features approach, where system operates on descriptors containing information about a locality of a particular graphical element. Local features give many possibilities for optimization for multi-pattern databases. In the application of retail product search we assumed that the detection rate, sufficient localization precision and low false detection rate are of the most importance. Computational efficiency is on the second place, as we do not assume real-time processing.

A multi-object detection system has to have a localization step, that may be used to divide the approaches into several groups. The first group may be a general object detection approach, which contain a saliency detector and a contextual image clustering. These methods are independent of any pattern and try to differ the background from foreground objects. There are some visual features, as edges and a frequency response, that can show areas of the image, that can be taken as an object. Another example of a general clustering approach has been presented in the work of Iwanowski *et al.* [12]. Unfortunately, this particular method fails in many scenes, as it needs very explicit shelves' and products' edges visible. Generic saliency methods failed in every one of the test photos, as scenes with products on shelves are salient in almost every spot. The second group of localization methods may use a voting scheme and local features. Local features in the scene can be matched against local features in the pattern. Consequent correspondences can be used to localize an object of a particular type in the scene. The complexity of such search can be minimized by using multiple detection stages, starting from the general search (logo or brand search) to a specialized identification (identification of the brand's member).

The last group of the localization approaches uses dense feature matching against a whole pattern database for each possible window in the scene. Algorithm has to generate a set of windows in any position and of any size. Such approach, called usually the sliding window approach, has been vastly used for object search purposes. Each window has to be processed as a standalone image in search for one instance of the object. It's obvious, that majority of the generated windows will not fit perfectly into object's envelope. The number of windows can be counted in thousands even in optimized window search. Each window has to be analyzed by a global image descriptor or a set of local descriptors. These descriptors have to be matched against the whole pattern database. The most advanced methods use hashing to minimize computational complexity in case of a large pattern database. Bag of words approach gives good results for local features as well. Despite of many optimizations in window search algorithms, such approach can be still too complex for modern machines in case of the analyzed applications. On the other hand, there are known well optimized multi-class multi-detection systems using modified HOG descriptors and LSH hashing methods. Unfortunately such methods use learning process and are not suitable for detecting specific, generic graphics with high amount of common visual elements. Many systems use global similarity metric, that gives good results in case of KNN (k nearest neighbours) queries. It's important though to create highly robust filter, that rejects false detections, as KNN queries don't provide information whether the best result can be accepted as a match. Simple distance thresholding may be not sufficient to accomplish this task effectively.

This paper presents the multi-detection system based on a method of vote space analysis. System is based on the invention shown in [15]. System uses local features and voting mechanism for localization and a cascade of filters to reject false detections presented in [16]. System is ready to use for a multiple stage detection and has linear scalability in regards to the pattern number. Using simple parameter automation mechanisms allowed maximization of detection rate. Achieving high amount of control over false detection response was the most important aspect of system's application. We used implementation of SIFT algorithm for tests, but presented approach can be used with any feature points containing scale and rotation information.

2 RELATED WORK

There are multiple works presenting building of a vote space for multi-detection purposes. Lowe in [19] proposes generalized Hough Transform for clustering the vote space. Authors of [2] create a 4D voting space and use combination of Hough, RANSAC and Least Squares Homography Estimation in order to detect and accept potential objects' instances. Zickler in *et al.* [28] use angle differences criterion in addition to RANSAC mechanism and a vote number threshold. Zickler *et al.* in [29] use a custom probabilistic model in addition to the Hough algorithm. Branch-and-bound approaches as in [27] are promising for multi-detection purposes in conjunction with Bag-of-words descriptors. Viola and Jones in [26] developed cascade of boosted features, that can efficiently detect multiple instances of the same object in one pass of the detection process. The method needs a time consuming, learning process on thousands of images. Method has been tested mostly on general objects, as people, cars, faces. Blaschko and Lampert in [6] use SVM to enhance sliding window process. Efficient subwindows search has been used in [17]. A most straightforward method of multi-detection is using all of the windows from sliding window algorithm, as used in Sarwas' and Skoneczny's work [24]. High effectiveness can be achieved with Histogram of Oriented Gradients [8] and Deformable Part Models [10]. Interesting use of DPM and LSH can be found in the work of Dean *et al.* [9]. The biggest drawback of the Deformable Part Models and Histogram of Oriented Gradients for analyzed application is that they usually need learning stage and are not rotation invariant.

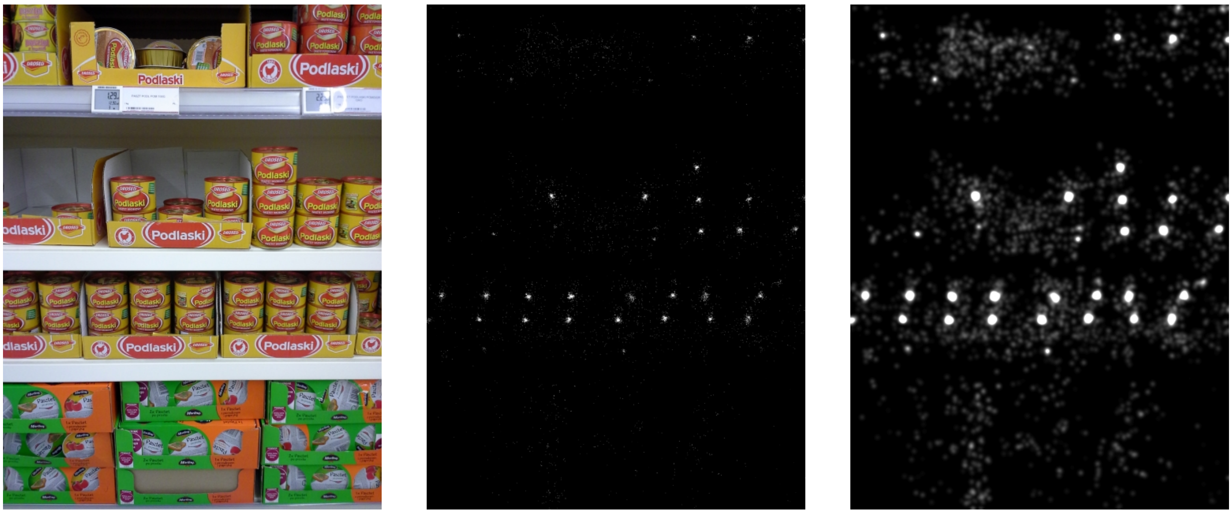
3 SYSTEM OVERVIEW

Presented system uses scale and rotation invariant local features for object detection. The core of the system is voting schema connected with a cascade of filters. Given a particular pattern we create the cascade of resized patterns. We extract local features in both the scene and the pattern images. Features from the two groups are matched against each other with a FLANN [22] algorithm. Correspondences are filtered with a contrast data and a color distance criteria. The threshold value for the contrast data distance is calculated as a middle value between the lowest and the highest distance values found in correspondence set. Color distance thresholding function does not apply for some values of HSL channels of a matched feature points pair. The contrast data distance is transformed to create the Adjacency value with a function:

$$adj(m) = 1 - \left(\frac{dist(m)}{thr}\right)^2, \qquad (1)$$

where m denotes the feature points match, *dist(m)* denotes distance between feature points in match m and *thr* is a distance threshold value.

Each correspondence is used as a vote in a multi-dimensional vote space. The vote space is not analyzed in a direct manner. It is projected onto the X, Y plane,



(a) Scene image.

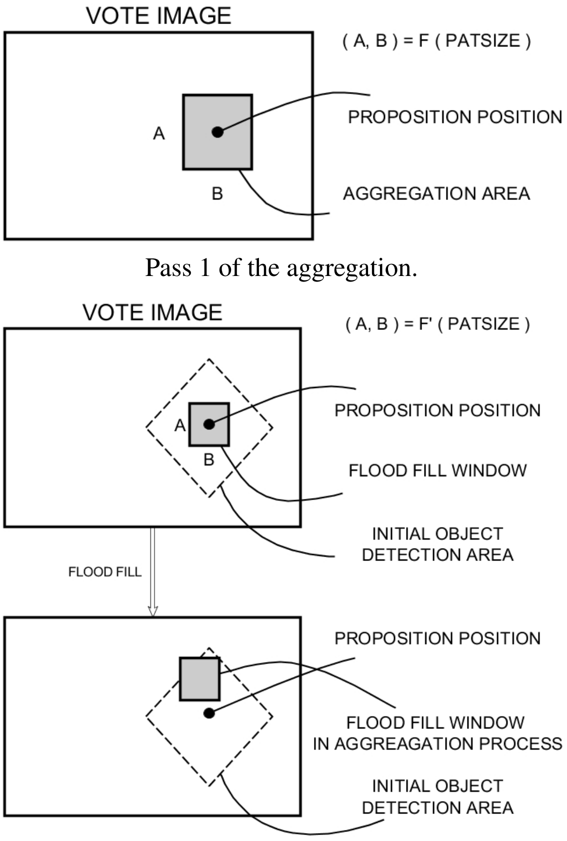
(b) Vote image.

(c) Blurred and normalized vote image.

Figure 1: Sample of a vote image generated while localizing a *Drosed* product.

where X and Y dimensions are identical to X and Y dimensions of the scene image. The adjacency values of each vote are summed for each (x, y) bucket and used as a cue to create a single channel image (called a vote image in this paper) of the same size as the scene. Adjacency values in the vote image can be normalized, and the image can be blurred to make it possible for human to analyze it and evaluate the efficiency of matching process. Such blurred and normalized vote image can be seen in Figure 1. Vote image is processed by a graphical local maxima detector. We found that Good Features To Track [25] works very well for this task. Local maxima in the vote image are further called propositions. Propositions are sorted by adjacency sum value in descending order. Each proposition is a center of a potential object instance in the image.

For each proposition, starting from the one with the highest adjacency sum, we perform a vote aggregation and a cascade filtering. Each of the filters in a cascade can accept or reject current vote aggregation process. Any rejection will lead to dropping the aggregation process and removing the processed proposition from the propositions sorted queue. Vote aggregation is a two-pass algorithm. Pass one of the aggregation collects all of the votes in a local area of proposition's position. After gathering of all of the votes in the local area, the unique filtering (discrabed later in this paper) is performed and the resulting group of votes is tested against a cascade of filters. In the second pass of the process the aggregation is conducted with the Flood Fill algorithm, starting from the proposition's position. The Flood Fill range is constrained by a scaled down object's envelope. Sizes of the local area in the pass one and of the Flood Fill search window in the pass two rely on a pattern size. The idea is presented in Figure 2. In pass two we've already got the estimation of the the object's envelope after analysis of the votes' data from the first pass. Second pass of the algorithm contains unique filtering and cascade filtering as well.



Pass 1 of the aggregation.

Pass 2 of the aggregation.

Figure 2: Aggregation process with aggregation window.

The unique filtering takes place after the vote aggregation and before the cascade filtering in each pass. It is a simple filter, which job is to make sure that only one correspondence is connected with each one of the pattern's features. It is important mechanism, that lowers the false positive detection rate.

Filters in a cascade can accept aggregated votes or reject them. Cascade consist of two types of filters. Vote data filters make use of data gathered in votes. Graphical filters use additional graphical data extracted from the scene image. Cascade filters comprise of: (1) vote count thresholding, (2) adjacency sum thresholding, (3) scale variance thresholding, (4) rotation variance thresholding, (5) feature points binary test, (6) global normalised luminance cross correlation thresholding. First pass of vote aggregation uses filters: (1), (2), (3) and (4). Second pass of the process uses filters: (3), (4), (5) and (6).

After successful vote aggregation and analysis, the object's occurrence is assumed. After that, all of the data corresponding to a detected object's area is erased from the vote image and the vote space. Then the next proposition can be analyzed.

The vote aggregation is the core of the detection system, but the whole framework is much bigger. The detection process for one product is performed in two phases. In phase one each pattern image is resized multiple times, till achieving minimal size. Each derivative pattern is processed as if it was an independent object's pattern. After detection process the occurrence consolidation is performed. It is likely, that the same products will be detected multiple times, as we generated couple of the same patterns but with different size. These detections are merged, and its adjacency sum values are summed. Each occurrence (detection) can be ranked on the basis of the adjacency sum value. The best occurrence is then chosen and a new pattern is extracted straight from the scene image. This new pattern is not resized. In phase two the detection process is performed for a second time for the extracted pattern. Final detections are consolidated and merged with detections from the previous phase.

After each product has been processed in a way presented earlier, the last consolidation is performed. It is likely, that some of the products in the scene will be detected as a different member of the same brand. Tested implementation doesn't use a multi-stage detection approach. We tested few different wrappings of the same product line to find out the basic detection resolution of the method. If two detections are overlapping, only the one with the best normalized adjacency sum is chosen. Normalization is performed for each pattern independently in regards to amount of visual features detected.

4 PATTERN VS OBJECT SIZE

The detection efficiency of the presented system depends on the assumption that, if the object exists in the specific area, then one has access to a substantial number of correct votes. We assume also, that the rest of the votes has noise-type distribution over scale, rotation and (x, y) location.

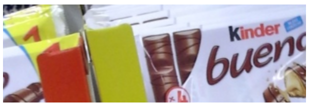
The system builds a cascade of patterns of different sizes from each one of the base patterns. For each pattern the detection process is performed. This method is a brutal approach, as the computation cost rises with the number of the resized copies. There are multiple benefits though.

All of the state-of-the-art local features lose repeatability characteristics for images with a different resolution. Matching capability of such features as SIFT or SURF decreases significantly, when the difference in size of the objects in the scene and pattern image are more than 2x. Similar results have been presented in works of Huynh *et al.* [11], Khan *et al.* [14] and Azad *et al.* [3]. The one reason for this is a fact, that the smaller image will usually has less detected feature points (assuming that both images have similar level of blurring). The average distance difference between two corresponding points in two images is getting bigger with the growth of resolution difference as well. This leads to increased contribution of false matches in the vote space.

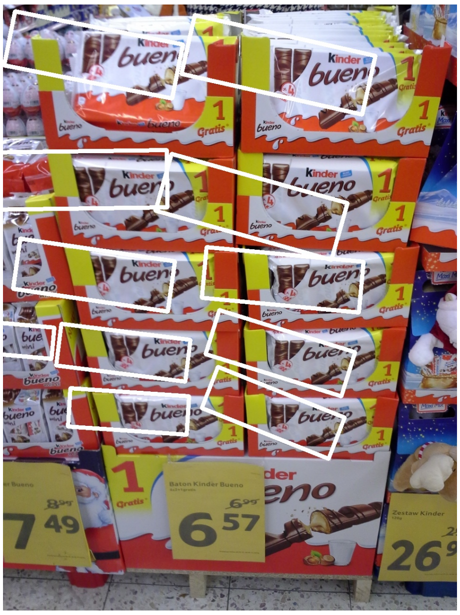
To overcome this limitations we decided to create sets of different pattern sizes. This is not only to minimize noise level or boost a matching capability. We can assume, that the number of features in the pattern is not far from the number of features extracted from the object in the scene image (if the object's sizes in the scene and the pattern images are similar). Additionally we can automatically reject all of the matches, which scale difference is over specified range. It's worth mentioning that, if there is a way to determine a real size of the scene frame through some kind of markings on the shelf, the pattern could be resized to the exact size of the object in scene (measured in pixels). This would accelerate processing greatly, yielding extremely low false matching, as the scale difference range could be narrowed down.

5 2-PHASE APPROACH

The 2-phase approach means using the new pattern, extracted from the scene image, for second phase of the detection process. After choosing the best detection in the first phase, we extract the exact area of the detection from the scene image to create a new pattern for second phase of detection. In the second phase there is no resize mechanism, as the new pattern has the exact size of the object in the scene. This mechanism increases computational complexity of the algorithm, but is the only mechanism in the test, that could achieve the highest possible detection rate. All of the modern visual features are susceptible to illumination changes, blurring, perspective warping, noise, bad color representation and many more characteristics of the natural photos. One of the simplest and the most straightforward way to overcome this limitations is to use the image, that is a part of the scene. This operation fits resolution, blur, lighting and noise conditions of the pattern to the conditions of the scene. In most cases, mentioned conditions are uniform for the whole scene. If we have the pattern extracted from the scene, the detection task becomes much easier, reaching even 100% detection rate for many scenes and objects. Unfortunately, it comes with a problem of false detections in scenes with no objects present. The best detection (a false detection in this case) could be chosen as a new pattern and, as a result, the system could identify the false occurrences in the scene in many other locations. This situation is shown in Figure 3. That's the reason for putting emphasis on lowering the false (negative) detection rate. The first phase does not need to detect multiple objects. It just needs to find one, real occurrence with high certainty. Presented system can be optimized to do such task, lowering the computation cost, as a result of processing only few of the strongest propositions.



False detection taken from phase one.



Multiple false detections after phase two.

Figure 3: Example of generation of multiple false detections after extracting the false detection from phase one. This example has been achieved by disabling filters in the cascade and using the pattern of object, that is not present in the image. Unfortunately such situation may occur with all filters enabled.

6 PARAMETERS

The system can function properly only, if its modules and processes are working jointly with the characteristics of the task. In practise it means many parameter adjustments before the system can be used in practise for broad problem characteristics. This chapter presents some of the main parameters, that must be considered while evaluating effectiveness of the system presented in the paper.

The first important parameter to determine is a scale factor for resizing the patterns in the first phase of object's detection. We used a scale factor of 2 (for each dimension) for this purpose. Resizing patterns allows narrowing down the scale quotient range in which we accept feature points matches as valid. We found that superimposing the scale acceptance ranges for different pattern sizes does not increase system's effectiveness in a meaningful way. The range for scale quotient has been set to (0.75, 1.5). Theoretically, the narrower the scale acceptance range, the less impact on detection has features' vulnerability for scale difference. Chosen parameters' values have been evaluated with test images and its further adjusting didn't yield any improvement in detections.

The distance threshold for filtering out the matches has been presented earlier in this paper, but it needs a comment. We decided to use half of the distance range, based on intuition and multiple tests, which did not showed any kind of strict correlation or mechanisms, which could lead to calculation of the ideal distance limit. It's mainly because the most reliable success rate metric can be extracted from the detection rate and false (positive) detection chance. Between detections and match filtering there are many other mechanisms that gain or lose its effectiveness with the distance threshold change. Filtering of matches is performed with use of a color filter as well. We reject point correspondences, which has the hue (in HSL color model) difference greater than 45. The filter works only, if the lightness (in HSL) is in range [10, 240] and the biggest difference in RGB channels for each point is over 10.

During proposition generation we use Good Features To Track algorithm, which has a scanning window parameter. The size of this parameter has big impact on the number of propositions detected and its accuracy. The bigger window can be interpreted as a blurring preprocessing of the vote image. The detector with too big window can generate inaccurate proposition's locations, which can compromise the aggregation of votes. A small window can generate too many propositions. The size of a scanning window in tests was calculated each time, as:

$$wSize(pSize) = \left(\left\lfloor \frac{pSize}{100} \right\rfloor + 1 \right) * 2 + 1, \qquad (2)$$

where *wSize* is a window size, and *pSize* is a bigger size of the (X, Y) dimensions of the pattern.

During the aggregation process votes are collected in a locality of the proposition. The locality is defined as a window of the same size, as during proposition generation. The Flood Fill algorithm in pass 2 has an aggregation window with the same size as well.

Each one of the filters in the cascade has its parameters. In the vote count thresholding we decided to process only groups of more than 6 votes. The adjacency sum thresholding makes a very similar kind of filter. The adjacency sum threshold is calculated as a number of feature points in the pattern divided by 200. This filter can reject groups of more than 6 votes but with a very weak adjacency values. It's important, that this filter is correlated with the pattern. In the scale variance thresholding we set the scale variance threshold for 60% of the average value of the scales squared in the aggregated set of votes. The rotation variance is tested in the same way as a scale variance. The difference lays in the calculation method of the rotation variance and average value. The calculation is not straightforward, because of the cyclical character of the rotation metric. The feature points binary test compares two binary vectors using Hamming distance. Two binary vectors of the same size are generated for an aggregated vote group - one on the pattern side, and one for the scene side. For each vote pair from the vote group two binary luminance tests are performed. Each test leads to a '1' value for L(p1) > L(p2) and '0' otherwise, where L() is a luminance returning operator, and p1 and p2 are the feature points from the scene (for first binary vector) or from the pattern (for second binary vector). When more than 25% of the bits are different between the vectors, we reject the vote aggregation. This test is not perfect, as many false detections have differences smaller than 25%. Nevertheless it can filter out huge amount of false detections, almost not affecting the positive detection rate, as positive aggregation yields very low distances in this test, especially for the phase 2 of the detection.

The global normalised luminance cross correlation thresholding is the last filter in the cascade. As it can accurately identify almost identical images, it is weak against different frame positioning and lightning conditions. Nevertheless it can filter out some false detections. Because we do not want to reject any positive detections we set the threshold to 0.5 for this algorithm (the cross correlation value's range must be normalised to *(0,1)*). In this filter each color channel is tested independently. The images are resized before the computation to a size of 20x20 pixels.

7 RESULTS

For experiments we used the same test database as in [16] for comparison. The image database consists of 120 shelf photos taken in 12 MPx resolution and scaled down to 3 MPx for testing purposes. The pattern group consists of 60 generic patterns of logos and product wrappings. Each shelf photo has been tested with each one of the patterns, conducting 7200 detection processes in total. Each scene contained very few classes of products, so most of the detection processes could generate only false positive detections. Average number of products presented in the scenes was *23.6*. Each pattern has been used with its original size, that was not higher than *700x700* pixels. The biggest ones led to generation of even three resized derivative patterns. Moreover the tests have been performed twice for scenes with the original 12 MPx and with reduced (3 MPx) resolution. The latter can be compared directly with the results of [16]. The feature points algorithm used for tests was the SIFT feature extractor and detector.

The testing database is strictly connected with the application of products search. During process of selecting photos for the database the scenes, where the shelf or the face of the products' front were rotated by more than 45 degrees from the photo's scene plane, were ignored. This selection was made manually. 45 degrees criterion gave a big field for error in this process. It is not a crucial problem, as in real application scenes with rotation bigger than 30 degrees can be marked as insufficient, if we want to achieve a detection rate above 90%. Database contains patterns, which show a whole front of the product as well as only a brand's logo. The patterns' framing have been chosen arbitrarily to test different approaches. Many scenes has very unfavorable lighting conditions and show multiple reflections on the products. Such scenes, connected with an imperfect or a very simple pattern, lead to poor detection rate. On the other hand, visually rich patterns lead to almost perfect detection rate, revealing even products, that are hard to notice for human.

The detection of a brand's logo is associated with a problem of putting the detections to a specific product's group. Some products of the same brand are very similar, with only slight local graphical differences. Presented system can detect a product, even if it is partially occluded. At the same time it can ignore the minor graphical difference and recognize the wrong member of the specific product's line. In real application such detections should be processed further to discriminate different variations of the product. One can use partial patterns with a bag-of-words approach on top of the presented aggregation method to do so. We call it a cascade detection process, where the thorough identification of the product is the result of many sequential algorithms.

In the Table 1 we showed global results for the tests. We achieved 89% detection rate for full resolution im-

Scene size	Detection Rate	False Detection Chance
12 MPx	89.0%	0.72%
3 MPx	84.4%	1.63%

Table 1: Detection rate and false (positive) detection chance for the tests.

Scene size	Average Number of False Detections
12 MPx	3,07
3 MPx	3,28

Table 2: Average number of false (positive) detections for a process, when the false (positive) detection occurred.

ages. At the 3 MPx resolution we achieved better detection than during tests in [16]. Higher resolution yielded lower chance for false detection. The interesting thing is, that in the test with lower resolution we achieved a false detection chance lower than in [16], even though the system makes few times more detection processes for different pattern sizes and because of a 2-phase approach. The reason for this result is the dynamic parametrization of the system. This parametrization couldn't prevent the rise in the overall number of false detections, that was more than 3 false detections per image (Table 2). This rise is connected with 2-phase approach, that uses the false detection as a new pattern, leading to a multiplication of the false detections.

8 CONCLUSIONS

Detection effectiveness of the system lays in three main aspects: proper vote group filtering, good parametrization, well defined pattern. The interesting thing is, that if we decide to use the filters described in this work and adjust the parameters, the pattern choice has the biggest impact on the performance. Size, sharpness levels, noise, lightning conditions - all of this characteristics can lower the detection rate even to 0% when chosen very unluckily. We found that the parameter-pattern dependencies and pattern extraction from the scene has the biggest impact on the system and should be researched much more. That is definitely a drawback of the one pattern approach, as the learning approaches tend to generalise the descriptor data to fit different application circumstances.

System shows promising results in tests. Simple approach to parameters adjustment and 2-phase processing improved detection ability of the system and is easy to analyze. System can achieve almost 90% detection rate with the false detection rate below 1%, that is acceptable in some real application.

In a future work we will optimize process of proposition acquisition to lower the computational complexity of the system. We are going to evaluate alternative visual features. We will evaluate possibility of using much smaller amount of visual features and a cascade approach to detection process.

ACKNOWLEDGEMENTS

This work was co-financed by the European Union within the European Regional Development Fund.

9 REFERENCES

[1] A Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517, June 2012.

[2] Pedram Azad, Tamim Asfour, and Rüdiger Dillmann. Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4275–4280, Oct 2009.

[3] Pedram Azad, Tamim Asfour, and RÃ¼diger Dillmann. Combining harris interest points and the sift descriptor for fast scale-invariant object recognition. In *IROS*, pages 4275–4280. IEEE, 2009.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin Heidelberg, 2006.

[6] MatthewB. Blaschko and ChristophH. Lampert. Learning to localize objects with structured output regression. In *Computer Vision - ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, pages 2–15. Springer Berlin Heidelberg, 2008.

[7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Computer Vision - ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer Berlin Heidelberg, 2010.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[9] Thomas Dean, Mark Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Proceedings of IEEE Conference on Computer*

Vision and Pattern Recognition, Washington, DC, USA, 2013.

[10] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, Sept 2010.

[11] Du Q. Huynh, Amritpal Saini, and Wei Liu. Evaluation of three local descriptors on low resolution images for robot navigation. In *Image and Vision Computing New Zealand, IVCNZ '09. 24th International Conference*, pages 113 – 118. IEEE, 2009.

[12] Marcin Iwanowski, Bartlomiej Zielinski, Grzegorz Sarwas, and Sebastian Stygar. Identification of products on shop-racks by morphological preprocessing and feature-based detection. In *Computer Vision and Graphics*, volume 8671 of *Lecture Notes in Computer Science*, pages 286–293. Springer International Publishing, 2014.

[13] Yan Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506–II–513 Vol.2, June 2004.

[14] Nabeel Younus Khan, Brendan McCane, and Geoff Wyvill. Sift and surf performance evaluation against various image deformations on benchmark dataset. In *DICTA*, pages 501–506. IEEE, 2011.

[15] Grzegorz Kurzejamski, Jacek Zawistowski, and Grzegorz Sarwas. Apparatus and method for multi-object detection in a digital image, September 2014. EU Patent 14461566.3.

[16] Grzegorz Kurzejamski, Jacek Zawistowski, and Grzegorz Sarwas. Robust method of vote aggregation and proposition verification for invariant local features. In *VISAPP 2015 - Proceedings of the Tenth International Conference on Computer Vision Theory and Applications*, March 2015.

[17] Christoph H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[18] S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, Nov 2011.

[19] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision* volume 60, pages 91–110. Kluwer Academic Publishers, 2004.

[20] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[21] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469, April 2009.

[22] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)*, pages 331–340. INSTICC Press, 2009.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571, Nov 2011.

[24] Grzegorz Sarwas and Sĺawomir Skoneczny. Object localization and detection using variance filter. In *Image Processing & Communications Challenges 6*, volume 313 of *Advances in Intelligent Systems and Computing*, pages 195–202. Springer International Publishing, 2015.

[25] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, Jun 1994.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518, 2001.

[27] T. Yeh, J.J. Lee, and T. Darrell. Fast concurrent object localization and recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 280–287, June 2009.

[28] Stefan Zickler and Alexei Efros. Detection of multiple deformable objects using PCA-SIFT. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1127–1132. AAAI Press, 2007.

[29] Stefan Zickler and Manuela M. Veloso. Detection and localization of multiple objects. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 20–25, Dec 2006.

3D Avatar for Automatic Synthesis of Signs for The Sign Languages

Diego Addan Gonçalves
Universidade Federal do Paraná - Brazil
diegoaddan@gmail.com

Eduardo Todt
Universidade Federal do Paraná - Brazil
todt@inf.ufpr.br

Laura Sanchez Garcia
Universidade Federal do Paraná - Brazil
laura@inf.ufpr.br

ABSTRACT

This paper discusses a synthesis system that generates, from a XML input representing gesture descriptors, a vector of configuration parameters that are executed by a 3D Avatar for use in the animation of Sign Languages. The development of virtual agents able to reproduce gestures of sign languages is very important to the deaf community, since in general they also have difficulties to read conventional texts. In this research project, a consistent combination of 3D editor Blender, CMarkup parser and graphics engine Irrlicht was used to develop a novel approach to sign synthesis, based on a recent XML model that describes hand gestures using shape, location, movement and orientation descriptors. The described experiments validate the proposed implementation model, which constitutes a promising alternative in the area of synthesis of signals for computational applications of Sign Languages.

Keywords

Gesture synthesis, HCI, Accessibility and usability, deaf community, graphics engine.

1 INTRODUCTION

Research and development of systems which use virtual agents for Sign Language interpretation and generation is still scarce, lacking investment and motivation to boost development in this direction. It is not broadly known that, in general, deaf subjects also have difficulties in the process of conventional text reading, due to the lack of meaning to the phonetic representation of syllables. An example of this is the social context where resources that facilitate the use of Sign Languages are fundamental but not currently available. For instance, environments such as hospitals, could have means to provide important instructions, such as emergency warnings, through a virtual agent, offering to deaf people the same speed in communicating information as provided by conventional text and voice alternatives.

The construction of a synthesis system between traditional written languages and Sign Languages depends on the definition of the inputs, such as video recognition of gestures or written symbols, and outputs, like a representation through a 3D avatar. In order to produce outputs in a generic and parametric way, it is necessary to have a base, or descriptive model, of the phonological aspects of Sign Language and also a system capable of interpreting these data.

If the data input is a video of captured gestures, the system can recognize the signals and then supply the data to the model that generates descriptive interpretation in the form of written characters representation or another descriptive representation. If the input data is in written character, or given by descriptive representation, the system can make the synthesis to the interpretation through a 3D avatar.

The existing tools to the visual representation of Sign Languages (Section II) have limited resources and do not generate movements automatically. In general the user has to drag a list of movements to the virtual agent to generate the desired animations, in a tediously and time consuming process.

These tools are based on complete and indivisible signs, being restricted to a very limited knowledge base.

Bearing in account the formal models presented in the literature, and the limitations of existing systems, this work aimed to contribute to the academic scenario in order to develop a service for automatic synthesis of signals of a Sign Language through a 3D avatar.

2 DESCRIPTION MODEL OF THE SIGN LANGUAGE LIBRAS AND EXISTING SYSTEMS

In this section, the necessary concepts related to computational models appropriate to describe the signs of Sign Languages and tools of textual interpretation are discussed, as well as other works related to computational systems for representation of Sign Languages using 3D Avatars.

2.1 Formal Representation of Sign Languages

The first visual architecture of schematic representation of signals, based on ASL, American Sign Language, was introduced in 1983, and it presented concepts of animation from a skeleton with movements based on formal modeling [1].

Since then, other studies have been developed adding new parameters, such as the type of motion, speed, repetition and symmetry of the arms [2] [3], and also using standard XML representation [4].

Recently a formal model that incorporates the parameters used in signals, structured through hierarchically organized classes, formally representing the Libras in an appropriated form oriented to computational use was developed [5].

This last model was chosen as the reference in the current work, although it was necessary an adaptation, described in the following sections, for use in 3D avatar. Figure 1 shows an example of some parameters and values of the formal model and its corresponding representation in an avatar.

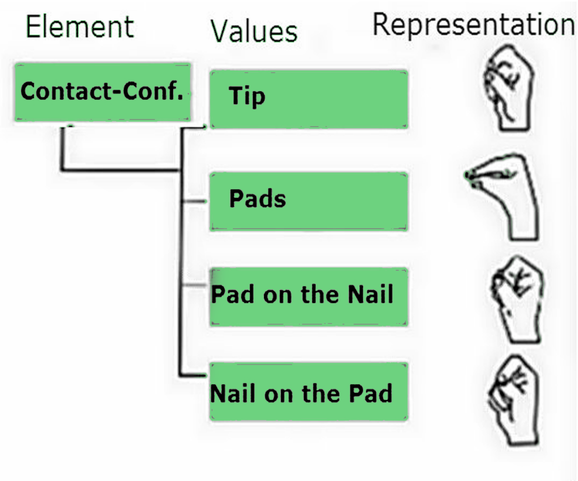


Figure 1: Here some contact configurations are show, where different finger positions indicate different types of contact of fingers.

2.2 Existing Tools for Representing Sign Language through 3D Avatars

From these formal models, and other forms of representation of Sign Languages, some software simulation of Sign Languages have been developed for commercial or academic purposes.

Gibet [6] was one of the first researchers to integrate the use of 3D for animation to a Sign Language. His work was based on a 3D arm with two joints. Two years later a early fingers configuration model was proposed, outlining some words, letter by letter, conceptualizing the idea of an animated hand that could be accessed over the internet using VRML [7].

Chadwick, Haumann and Parent [8] presented a model for creating an animatable body, human or not, based on three layers: skeleton, muscles and skin. The main commercial computer graphics tools work with a system like this, since only the layer of muscle is not indispensable to obtain a full animatable model based on deformation of polygons.

Research for an outline application for the synthesis of a Sign Language is a constant concern, expanding its outputs to the web. In [9] a model is presented based on two main classes, the client and server. The first consists of a common web browser, with support for technologies such as WebGL (HTML5 API that allows for the rendering of 3D graphics in web browsers) or O3D (open source API for creating 3D applications in web browsers) and Java Script (scripting language for web browsers). The server class receives requests from the client side and does the communication and translation animation, followed by conversion needed to pass the results to the browser on the client side.

Among these systems, it is relevant to quote those from VCom Gesture Builder [10], Max's Einfach Teilhaben [11] and Sign 4 Me [12], which propose the representation of Sign Languages through a 3D Avatar.

Yi, Harris and Descalu [13] also suggest that a major problem in these systems is the fact that users must know the native language of the software. The authors also reinforce the idea that the evaluated systems don't have a simple and intuitive interface, hindering the interaction of beginners. An important point in which the authors focus is that these tools may not have extended their content being limited to a set of pre-defined phrases or words. In other words, these systems rely on user controlled content, not being automatically animation software.

We conclude that the main limitations of these existing systems arise from the characteristic that they didn't use parameters based on a formal model, and so the animations are indivisible full signs, not generated automatically. Also they spell letter by letter a syllabic representation of the signal meaning, assuming that the deaf user has the necessary knowledge of a traditional written language in order to understand the gesture output. Moreover, most of these software are commercial, being neither freely distributed nor open source.

Thus, one attractive alternative to overcome these limitations is to develop a system that makes the synthesis of the input parameters based on a formal model of computing Sign Language, producing an output through a 3D avatar, as shown in Fig.2, where the synthesis process is the most important stage.

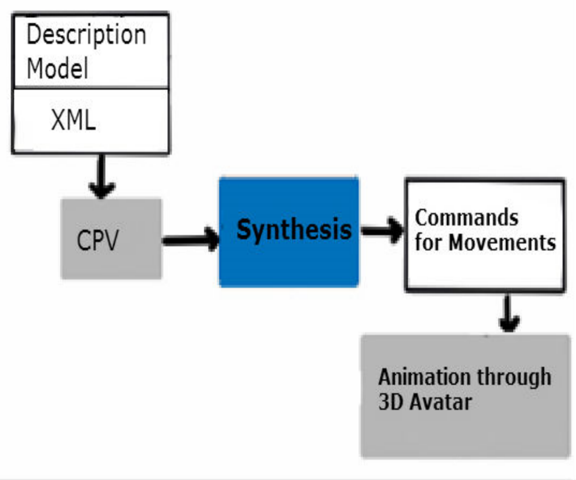


Figure 2: Block diagram of the proposed system, where CPV means Configuration Parameters Vector (Described in Section IV), that represents sequences of the required poses for each joint of the avatar's skeleton.

3 DEVELOPMENT OF 3D AVATAR AND ANIMATION SYSTEM

To implement the synthesis process it was first necessary to create a 3D humanoid object concerning the Avatar 3D. Its construction involved the following steps: design of the avatar mesh, research of avatar modeling techniques and choice of an appropriate option, definition of the necessary articulation points, and rigging construction. The last one is the structure used in the animation process and its connection to the mesh defines the reactions in the mesh to movements of the controller structure connected to it.

The modeling technique used was the poly-by-poly, following the concept art that the mesh was constructed from polygon extrudes [14]. At the end of this process an object was obtained with 2,276 polygons, which may be considered a low-count poly object. Fig.3 shows the concept art and the corresponding 3D Avatar mesh that was built using this technique.

In the following, the animation structure and rigging process for the 3D avatar was built. This structure includes controllers for the hands, arms, shoulders, torso and head. Facial expressions were not yet integrated at this point, but will be explored in future work.

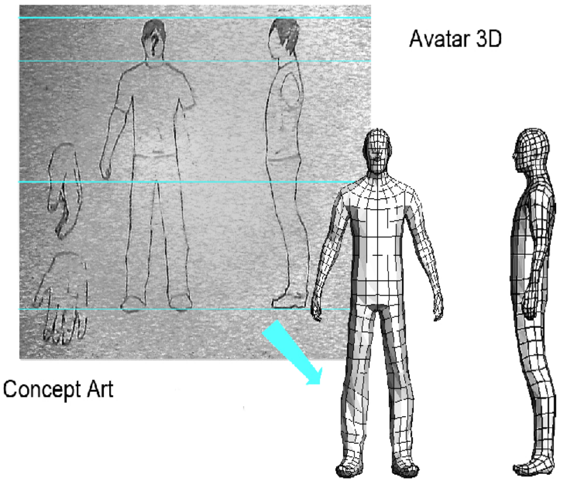


Figure 3: Concept Art and 3D Avatar.

The structure of animation controllers was integrated with the mesh through a technique where a weight is applied at each vertex of the structure relative to a particular controller.

Then UVW (map coordinate technique) mapping of the mesh was performed, where 3D structure is unwrapped into a 2D plane, enabling the application of the texture material needed to give to the avatar is superficial finishing [14].

With that, a texture was applied in the surface of the object for visual effects, finalizing the process of creating the 3D avatar. The most important point in the 3D avatar construction, beyond the concern to keep the mesh with few polygons, was to adjust both the physical model (mesh), and the animation framework, with the parametric model applied to the system.

The structure has a particular emphasis on hands, which were the most used in the tests and are the attribute of greatest importance in the formal model. In the future it is important to consider the Facial Expressions, which were not detailed in the formal model [5]. For the time being the implemented parameters were the hands and arms. Thus, at the rigging these points were considered to beof greater importance in terms of realism in the distribution of the weights of the vertexes and the spatial changes (collision with other parts).

4 AUTOMATIC SYNTHESIS OF SIGNS

This section describes the behaviour of the proposed system for the automatic synthesis system, generating the transcription to an output of hand and arms elementary movements through a virtual agent.

4.1 CPV - Configuration Parameters Vector

The descriptive model of Libras for computational use is divided into elements and sub-elements. The hand, for example, is considered as an element in formal model and the configuration of the hand is a sub-element with their respective values, like spatial coordinates or hand configuration. These parameters are represented by a vector of configuration parameters, called Configuration Parameters Vector (CPV).

This CPV is based on the formal model elements and are used XML external files to organize and register them, descriptively, separating the parameters by hierarchical packages that have elements and sub-elements and their values.

Set the schema formal model [5], a data structure based on tree was built to organize the elements used in the following test. This tree retains the hierarchical condition of the elements, and its construction prioritized elements concerning the arms and hand expressions.

The hand settings values, considering the Brazilian Sign Language Libras, have 61 possible values, and therefore it was decided that the rotation control and direction of the palm are the variable values of the application.

This representation differs from the formal model that suggests to control each finger for each motion value. Including the fingers would bring an unnecessary computational cost, so was defined use a "hand configuration" as elements. Was used for the tests the Libras hand settings and values for all other parameters defined in the model, such as motion, contact, direction of rotation, arm position and palm position.

Fig.4 shows an example of the created tree, on an AVl tree model. This tree graphic refers to the nodes of the sub-tree of non-manual expressions, where each node represents a value in the struct, hierarchy-dependent, based on formal parametric model. These sub-trees was expanded adding their specific parameters of articulations, movements and directions. Was built one sub-tree for each parameter and their elements.

This data structure was used for the construction of the CPV inputs. The interesting thing in CPV input format used in this work is that, since the entries are organized hierarchically like a tree, it is possible in the future use these algorithms for path decision making and optimization by the nodes.

Since an input is obtained, the system read the input file and recognize its structure, elements and values. The elements are the nodes that make up the chain and their values are the positions that each element of the CPV should take.

Fig.5 illustrates how this initial process works.

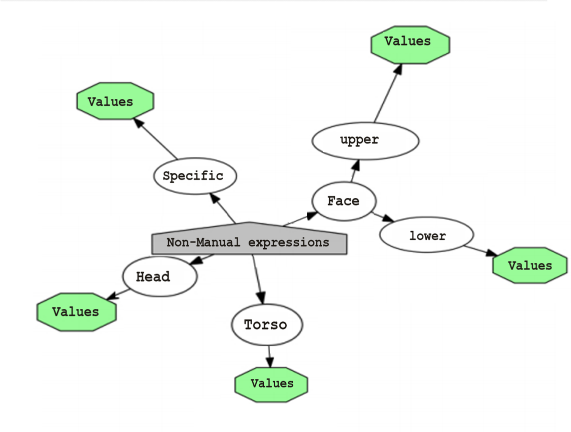
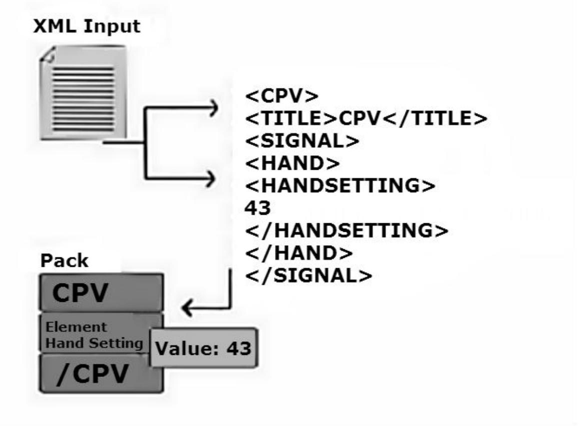


Figure 4: Sub-tree of CPV, hierarchically organized.



```
<CPV>
<TITLE>CPV</TITLE>
<SIGNAL>
<HAND>
<HANDSETTING>
43
</HANDSETTING>
</HAND>
</SIGNAL>
```

Figure 5: CPV interpretation.

We then conducted integration tests, that in principle were independent of the use of graphics engine (tool required for the next stage of the work) between the main application in C++ and the input file for the CPV. The test procedure was to create a function in the main application that can read the XML input file, extract and recognize their data, and pass the information to the application variables that can be used in the process or run time.

The data in test file were divided into two packs separated by tags, the first was called "element1" with the value "1", and a second pack called "element2" with the value "2". Basically the system should read this file by the application, separate the packs properly in hierarchical order and identify their values.

These tests were conducted using the parser library CMarkup that allows navigating XML files. With the system being able to read and identify elements of the input, the next procedure was the implementation in the graphics engine, where synthesis of the obtained data and generation of the output through the 3D avatar happens.

It is important to emphasize that the application of the formal model parameters used in virtual reality brings a new and promising scenario for the sign representation of Sign Language systems, correcting the main conceptual problem of this system is that the construction of phrases from a principle without unity, dealing with sentences like elements and not independent letters.

4.1.1 Automatic synthesis through the 3D avatar

The process of synthesis from data found in XML input to an output to Avatar 3D, is based on the 3D editor Blender, the parser CMarkup and graphics engine Irrlicht. The system software was implemented in C ++.

In order to build the animation and the corresponding call parameters, two paths can be taken. The first method is to make the code structure generate the animation movements of the graphics engine making each call associated to a pivot point bones. In this sense, each bone must be moved following the hierarchical structure of formal model, descending from the main element to the last sub-element chain. The second method are moving the mesh of 3D Avatar using the weight of each vertex, through an skeleton structure based in curves.

The system code indicates which bone will be moved and its coordinate position (in X, Y, Z environment position). After moving the first element of the chain of CPV it performs the same procedure on the following chain element (child node), and repeats the procedure until the last element in the chain is achieved.

In this process, the system only needs the textured Avatar 3D with the animation structure rigged, all calculations and procedures are executed from the main system through the graphic engine, which control animation structures.

Fig.6 shows the operation of this method.

A full test of the system was performed, integrating the parser to the environment with the graphics engine and using the methods to control the animation structure, in order to identify the input file and the information contained therein.

For the experiments, an entry with two elements was used, one representing an arm movement and the other specifying a hand configuration, one of the 61 standard possibilities from hand in Libras [15].

The mapping of these two configuration was done in the 3D editor, defining chains of sequential frames, using a morph modifier, for each block, by calculating the spatial alteration of each point of controller structure.

After recognizing the elements blocks for the CPV, each value was passed to a variable in the main application, so it could be used on the application process. Then a comparison with the values extracted was performed and, then, the application recognized the input block, searched the database and executed the mapped elements, exported to the mesh. These comparisons follows the hierarchical sequence of formal model.

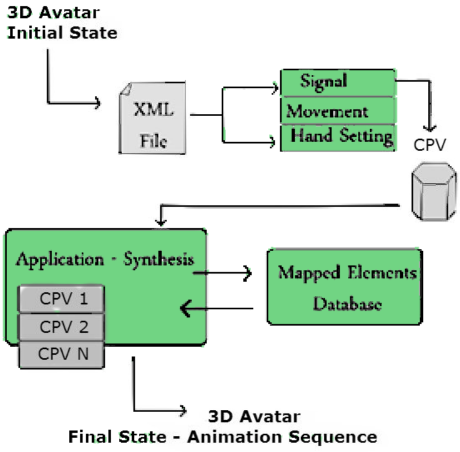


Figure 6: Synthesis Process. Begins by parse, extract the data configuration parameters, and applies for coordinates in the virtual environment rendering through 3D Avatar with the mapped movements.

In the following there is an slice of pseudo-code, concerning the recognition process of the input elements and passing parameters to the output generation process in the graphics engine Algorithm 1.

```
Data: XML Input
Result: Output in Graphic Engine - synthesis through
        3D avatar.
while (XML.Element = nextParameter AND
VariableRoot.element != Null) do
    read current element;
    if GetChildData == avatarParameter then
        gets coordinates;
        avatar = nodeSetFrameLoop(Value);
        counter = nextPointer;
        readNextParameter(counter)
    else
        counter = nextPointer;
        readNextParameter(counter);
    end
end
```

Algorithm 1: Read XML Parameters

Therefore, it could be verified that the output generated by 3D Avatar followed exactly the order described in the XML input. Fig.7 shows the output of 3D Avatar animation using the method described.

5 VALIDATION

For validation purposes a new test was conducted, running the entire process. The sign for the word "motorcycle" representation in Libras was chosen to be applied

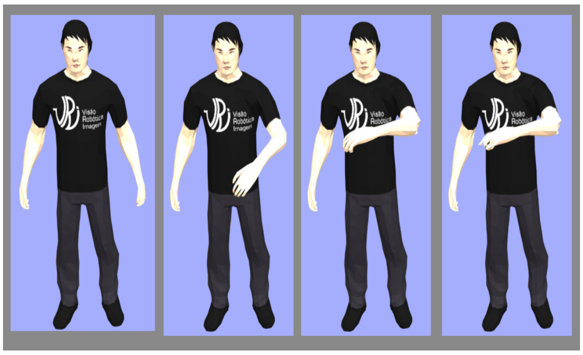


Figure 7: Animation Output.

to the complete synthesis process. As the test conducted in the previous section, the initial step was to build the XML input regarding the formal model blocks.

The validation protocol aimed testing the extraction of inputs through external XML file, evaluating the mapping of individual elements as well as the deformation in the 3D mesh.

Then we carried out the process of mapping the elements defining what will be the representation of a particular bone, or bones, related to a specific input. The bones constitute the articulated skeleton structure of the avatar.

For the "motorcycle" signal, the following elements were defined for the CPV: symmetry between the bones of the forearm and hand, bone turnover regarding the forearm at 90 degrees, hand setting, rotating the wrist of the right hand 90 degrees, repeating 3 times the last element.

Then we performed the synthesis process, where the description has been read from the XML file in order to find the CPV elements described earlier, and the system could find all blocks of configuration elements in hierarchical sequence. Since the call order and the elements were mapped, the system generated the corresponding output through 3D avatar as shown in Fig.8.

A test was built with two additional configuration parameters related to hand position without symmetry, besides rotation and movement of the arm. The tests followed, in general, to analyze the process unit from reading the CPV to exit through the graphic engine. In all cases the two methods used were shown in previous section, in order to assess behavioral differences between the different forms of the mesh control.

All tests presented in this paper were performed on a machine with the following settings: AMD Turion II X2, video card ATI Radeon HD 5470, 4 GB DDR3 memory.

As for performance and run time, all tests were processed in less than 1 second, since that technique, although involving 3D graphics, that usually constitute a heavy process, uses no complex lighting system, nor physical effects as secondary objects, scenarios or meshes which are not part of the 3D Avatar, which could make the rendering process heavy.

Concerning the results, the second method of synthesis process presented in the previous section, which uses a database of mapped elements in Blender itself, proved to be more advantageous than the first one, as discussed bellow.

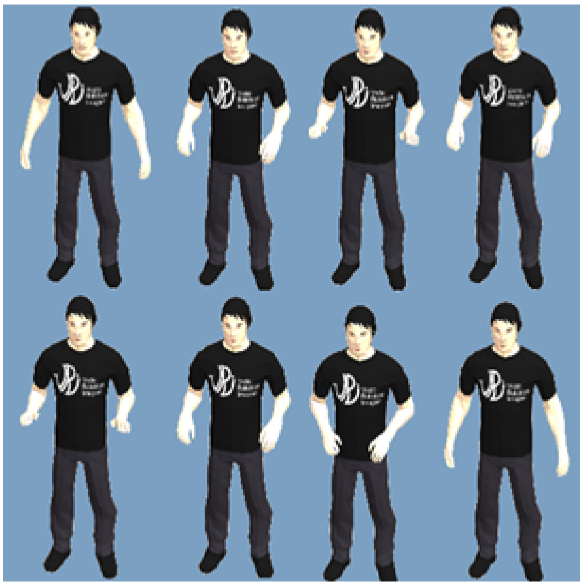


Figure 8: Motorcycle Animation Output.

The first method of testing the avatar 3D, in some cases initiated in a false position, which reflected across the output animation. Furthermore, since in this case the mapping process and calculations of positions happens at run time, the rate of frames per second in some tests fell by cutting approximately one frame to a sequence of forty.

In the second case, most of the processing is done during the mapping of the elements in 3D editor, and the processing of each block in the synthesis process, occur by calling blocks of pre-calculated position frames, which secure a less processing load.

This difference in the two methods proves valuable in the future, where the system has to calculate not only a sign, or part of it, but a entire conversation.

Still, the use of graphics engine lets you view, in the near future, the integration of these methods to the development of concepts for mobile devices or web browsers. Thus the spread and use of this kind of system achieves a significant portion of users, deaf or not, making this type of application, natural part of current systems.

6 CONCLUSIONS

The main contribution of this work is the development of a synthesis system through a 3D avatar for use in automatic Sign Languages animation, based on a formal computational parametric model. With further work, this system can be used by the deaf community in fact, already working with elements of ACP, and not ready with signs or spelling, solving the problems of existing systems.

This formal model provides that the terms and words are represented within a real context and use of the deaf community, unlike the known systems. The traditional way, by spelling, generates an animation without unity, and difficult to understand. This step is important for the development of computational systems of representation in virtual reality facing the deaf community start of a right principle, using a formally input accepts.

The general contributions of this work are the application of conceptual formal model [5] in a real virtual reality environment, and the developed algorithm to extract the CPV information and translate into coordinate information for the animation of the 3D avatar. As a restriction, in this moment, can be cited especially the process of mapping entries to frames of position of animation, as well as inclusion in the model of facial expressions and improvements in rigging animation.

Facial animations and non-manual movements are not yet implemented in the formal model, however with the results of this work can be defined as building these new parameters following the same principles developed for the construction of the system to the arm and hands.

7 REFERENCES

[1] J. Loomis, H. Poizner, U. Bellugi, A. Blakemore, and J. Hollerbach, "Computer graphic modeling of american sign language," *SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 105–114, Jul. 1983. [Online]. Available: http://doi.acm.org/10.1145/964967.801139

[2] R. Quadros and L. Karnopp, "Língua de sinais brasileira: Estudos linguísticos," *Artmed*, vol. 1, 2007.

[3] W. Stokoe, "Sign language structure: An outline of the visual communication systems of the American deaf," *JOURNAL OF DEAF STUDIES AND DEAF EDUCATION*, vol. 10, no. 1, pp. 3–37, WIN 2005.

[4] H. Sagawa and M. Takeuchi, "A teaching system of japanese sign language using sign language recognition and generation," in *Proceedings of the Tenth ACM International Conference on Multimedia*, ser. MULTIMEDIA '02. New York, NY, USA: ACM, 2002, pp. 137–145. [Online]. Available: http://doi.acm.org/10.1145/641007.641035

[5] D. Antunes, C. Guimaraes, L. Garcia, L. Oliveira, and S. Fernandes, "A framework to support development of sign language human-computer interaction: Building tools for effective information access and inclusion of the deaf," in *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*, May 2011, pp. 1–12.

[6] S. Gibet, "Synthesis of sign language gestures," in *Conference Companion on Human Factors in Computing Systems*, ser. CHI '94. New York, NY, USA: ACM, 1994, pp. 311–312. [Online]. Available: http://doi.acm.org/10.1145/259963.260372

[7] S. Geitz, T. Hanson, and S. Maher, "Computer generated 3-dimensional models of manual alphabet handshapes for the world wide web," in *Proceedings of the Second Annual ACM Conference on Assistive Technologies*, ser. Assets '96. New York, NY, USA: ACM, 1996, pp. 27–31. [Online]. Available: http://doi.acm.org/10.1145/228347.228353

[8] J. E. Chadwick, D. R. Haumann, and R. E. Parent, "Layered construction for deformable animated characters," *SIGGRAPH Comput. Graph.*, vol. 23, no. 3, pp. 243–252, Jul. 1989. [Online]. Available: http://doi.acm.org/10.1145/74334.74358

[9] M. Hrúz, P. Campr, Z. Krňoul, M. Železný, O. Aran, and P. Santemiz, "Multi-modal dialogue system with sign language capabilities," in *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS '11. New York, NY, USA: ACM, 2011, pp. 265–266. [Online]. Available: http://doi.acm.org/10.1145/2049536.2049599

[10] "Gesture Builder software," http://www.vcom3d.com/?id=gesturebuilder, accessed: 2015-05-05.

[11] "Max E.T. software," http://www.einfach-teilhaben.de/DE/StdS/Home/stds_node.html, accessed: 2015-05-05.

[12] " Sign 4 Me software," http://www.vcom3d.com/, accessed: 2015-05-05.

[13] B. Yi, F. C. Harris, Jr., and S. M. Dascalu, "From creating virtual gestures to "writing" in sign languages," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '05. New York, NY, USA: ACM, 2005, pp. 1885–1888. [Online]. Available: http://doi.acm.org/10.1145/1056808.1057047

[14] K. L. Murdock, *3ds Max 2012 Bible*. Wiley Publishing, 2011.

[15] A. Porfirio, K. Lais Wiggers, L. Oliveira, and D. Weingaertner, "Libras sign language hand configuration recognition based on 3d meshes," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 1588–1593.

Applying Filters to Repeating Motion based Trajectories for Video Classification

Kahraman Ayyildiz, Stefan Conrad
Department of Computer Sciences
Heinrich Heine University Duesseldorf
Universitätsstraße 1, 40225 Duesseldorf, Germany
kahraman.ayyildiz, stefan.conrad@uni-duesseldorf.de

ABSTRACT

The presented video classification system is based on the trajectory of repeating motion in video scenes. Further on this trajectory has a certain direction and velocity at each time frame. As the position, direction and velocity of the motion trajectory evolve in time, we consider these as motion functions. Later on we transform these functions by FFT and receive frequency domains, which then represent the frequencies of repeating motion. Moreover these frequencies serve as features during classification phase. Our current work focuses on filtering the functions based on the motion's trajectory in order to reduce noise and emphasize significant parts.

Keywords

Action Recognition, Video Classification, Repeating Motion, Frequency Feature, Filter, Occlusion

1 INTRODUCTION

Today there is a strong demand for computer vision research, since recognition and tracking of objects or motions are core subjects for some major industries. Face tracking for videoconferencing, computer controlling by gestures, size measurement of components on band conveyors or positioning of industrial robots are only some examples, where computer vision has already been established successfully. Moreover computer vision is also needed when it comes to video annotation and classification for video databases.

Current research work brings action recognition and classification by repeating motion into focus. In [AC2012] we already presented the basic idea of our approach. Now we extend our system by adding different filters in order to smoothen or to emphasize repeating motion in videos. Hence the experimental phase is concerned with accuracy and runtime analysis for different filters. Especially when recording conditions for videos differ, filters can compensate these differences. This pertains for varying illumination, resolution, occlusion, shaking or angle.

The analyzed filters in the experimental part of this research work are applied to repeating motion based trajectories. These trajectories serve as the basis for feature extraction. In the field of motion analysis filtering is sparsely researched. Thus our contribution at hand points out the effect of filters on motion trajectories and resulting features.

2 RELATED WORK

Videos can contain key-frames, texts, audio signals, motions or meta-data. Hence video classification can be realized in various ways. In our research work we focus on repeating motion, which is also discussed in a similar way by [MLH2006] and [CCK2004]. [MLH2006] deals with repeating motion of human body parts tracked by Moving Light Displays (MLD). Frequency peaks of Fourier transformed MLD curves are considered as features of repeating motion. In [CCK2004] Cheng et al. analyze sports videos by using a neural network based classifier. They receive two main frequencies for each video by transforming series of vertical and horizontal pixel motion vectors. The transformation takes place by a modified fast Fourier transform. Furthermore the authors of [FZP2005] propose a hybrid model for human action recognition, which is robust against occlusion. This model is based on position, velocity and appearance of body parts.

The filters we consider in this work are particularly applied in image processing and hardly in video analysis. Research in [VUE2010] and [MAS1985] shows that the Lee filter performs better than the average or median filter when it comes to noise reduction for images. Alsultanny and Shilbayeh analyze a series of filters by applying them to satellite images [AS2001]. Here median, average and low-pass filters lead to similar results. Concerning edge detection filters

the so-called *Prewitt filter* works more accurate than Laplace filter.
In the field of video content and motion trajectory analysis there is sparse research done on the application of filters.

3 APPLICATION OVERVIEW

The flow diagram in figure 1 illustrates the different phases of our system [AC2012]. It starts with video data input containing repeating motions as painting, hammering or planing for instance (home improvement). Next regions with motion are detected for each clip frame by frame. For region detection the color difference of pixels in two sequential frames is measured. On the basis of motion regions we calculate image moments. We consider the chronological order of image moments as a *1D-function*, which again represents the main motion in a video sequence. This 1D-function is filtered in order to remove noise respectively to weight important parts. Moreover the result is transformed and we receive a frequency domain describing the frequencies of repeating motion in the video. By dividing the frequency axis into intervals of same length, average amplitudes for each interval are calculated. We name these averages *Average Amplitudes of Frequency Intervals* and refer to them as *AAFIs*. AAFIs set up the final feature vector for each video. At last a radius based classifier (RBC) utilizes this feature vector for the purpose of computing the nearest class for a video.

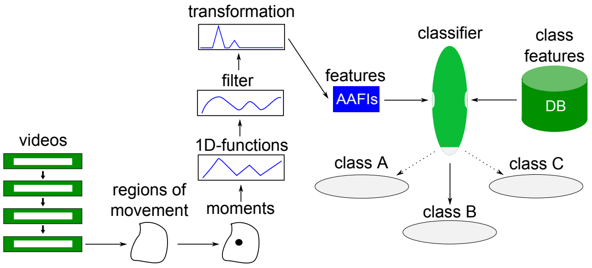


Figure 1: Flow diagram of the whole classification process

4 IMAGE MOMENTS AND 1D-FUNCTIONS

Once motion areas in a video scene are detected image moments can be determined. These image moments lead to 1D-functions, which are explained and defined formally in this section.

Regions of Motion

Figure 2 shows a person painting a wall. We detect regions with movement by comparing two sequential frames of this activity. Further on we measure color differences between these two frames for each pixel. The color difference of a pixel exceeding a predefined threshold combined with a minimum number of neighbor pixels with a color difference beyond the same threshold defines a pixel to be part of a movement. Thus a region with motion is represented by the entirety of pixels with motion. Pixel differences of the two frames shown in figure 2 point out regions with movement, which again are visualized by a monochrome image on the right. It is obvious that the most active areas are the paint roller, the hand, the forearm and the upper arm. Therefore the centroid of regions with motion follows exactly the right forearm. As a result the painting activity sets a specific motion trajectory.

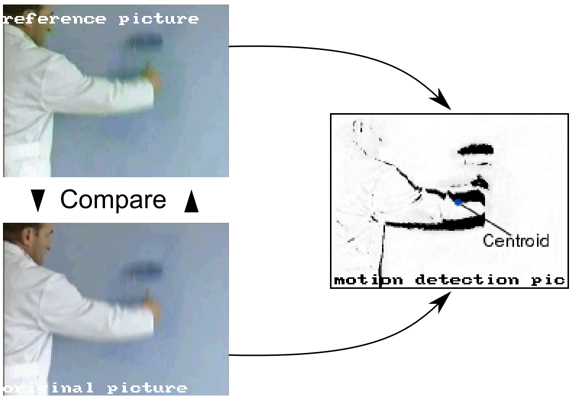


Figure 2: Regions with pixel activity and centroid

Image Moments

An image moment is defined as an image's weighted average of pixel intensities. It can describe the bias, the area or the centroid of segmented image areas. The two main image moment types are raw moments and central moments. Raw moments are sensitive to translation, whereas central moments are translation invariant. The next equation defines a raw moment M_{ij} for a two dimensional monochrome image $b(x,y)$ with $i,j \in \mathbb{N}$ [WSL1995]:

$$M_{ij} = \sum_x \sum_y x^i \cdot y^j \cdot b(x,y) \qquad (1)$$

The order of M_{ij} is always $(i+j)$. M_{00} is the area of segmented parts. Consequently $(\bar{x},\bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00})$ determines the centroid of segmented parts.

Deriving 1D-functions

Video frames have a chronological order. Hence a series of moment values is also depending on time t. Now we define a 1D-function $f(t)$ as a series of these moment values by considering only one dimension. For centroid coordinates $(\bar{x}_t,\bar{y}_t) = (M_{10_t}/M_{00_t}, M_{01_t}/M_{00_t})$ we decompose function $f_c(t) = (\bar{x}_t,\bar{y}_t)$:

$$f_{c_x}(t) = \bar{x}_t,\ f_{c_y}(t) = \bar{y}_t \qquad (2)$$

Experiments in section 6 use only $f_{c_x}(t)$ and $f_{c_y}(t)$ instead of $f_c(t)$, because the 1D-function transforms result in more accurate frequency domains than 2D-function transforms. By equation (3) we define the direction of an image moment at time t for any 1D-function $f(t)$.

$$f_d(t) = \begin{cases} +1, & \texttt{if } f(t) - f(t-1) > 0 \\ 0, & \texttt{if } f(t) - f(t-1) = 0 \\ -1, & \texttt{if } f(t) - f(t-1) < 0 \end{cases} \quad (3)$$

Now the speed of an image moment at time t is defined as follows:

$$f_s(t) = |f(t) - f(t-1)| \quad (4)$$

5 FILTERS FOR 1D-FUNCTIONS

In real world videos motions of the same activity are never exactly the same and motion trajectories differ from ideal mathematical functions. Unexpected motions, occluded motion or low recording quality can reduce the clarity of 1D-functions and therefore the system's accuracy. In order to improve the clarity various filters can be applied. Filters can reduce noise, smoothen trajectories or emphasize edges, which mean the change of direction in the case of 1D-functions.

Maximum Filter

A maximum filter substitutes each value of a data sequence by a maximum value inside a predefined radius. Let sequence (a_i) with $a_i \in \mathbb{N}$, $i = 0, \ldots, n$ and let radius $r \in \mathbb{N}$. Further on we define $N_r(i)$ as the set of neighborhood indices of sequence element a_i:

$$N_r(i) = \{x \mid 0 \le x \le n \wedge i - r \le x \le i + r\} \quad (5)$$

By these definitions we can compute the maximum value around a_i:

$$max_r(a_i) = \max_{x \in N_r(i)} a_x \quad (6)$$

Now applying the maximum filter the new sequence $(q_{i_{max}})$ gives:

$$(q_{i_{max}}) = (max_r(a_0), max_r(a_1), \ldots, max_r(a_n)) \quad (7)$$

Median Filter

The median filter substitutes each value of a sequence by a medium value inside a given radius. Again we consider sequence (a_i) with $a_i \in \mathbb{N}$ and $i = 0, \ldots, n$, radius $r \in \mathbb{N}$ and $N_r(i)$. For each value a_i we compute a sorted subsequence $(s_j) = (s_1, s_2, \ldots, s_m)$ inside radius r, where again $N_r(i)$ determines the indices neighborhood. For m as the length of (s_j) we define:

$$med_r(a_i) = \begin{cases} \frac{1}{2}\left(s_{\frac{m}{2}} + s_{\frac{m}{2}+1}\right), & \texttt{if m even} \\ s_{\frac{m+1}{2}}, & \texttt{if m odd} \end{cases} \quad (8)$$

For (a_i) the usage of a median filter results in $(q_{i_{med}})$:

$$(q_{i_{med}}) = (med_r(a_0), med_r(a_1), \ldots, med_r(a_n)) \quad (9)$$

Average Filter

By applying the average filter each value of a sequence is replaced by the average of all values inside radius $r \in \mathbb{N}$. For sequence (a_i) and $N_r(i)$ as the indices neighborhood we replace each value a_i as follows:

$$avg_r(a_i) = \frac{\sum_{x \in N_r(i)} a_x}{|N_r(i)|} \quad (10)$$

Hence we formulate sequence $(q_{i_{avg}})$ as:

$$(q_{i_{avg}}) = (avg_r(a_0), avg_r(a_1), \ldots, avg_r(a_n)) \quad (11)$$

Lee Filter

J. S. Lee proposes a statistical filter for digital images [LEE1980]. Lee assumes that each image contains natural noise, which can be removed pixelwise. Let σ^2 the variance inside radius r, δ a predefined noise energy and $\sigma^2 < \delta$, then a pixel is replaced by the average inside r. For $\sigma^2 > \delta$ the original value is replaced by another functional value: A high variance σ^2 means that the original value stays almost the same, because it is significant. Lee's filter can also be applied to 1D-functions. For sequence (a_i), radius r and $\beta = max(\frac{\sigma^2 - \delta}{\sigma^2}, 0)$ with $\beta \in \mathbb{R}^+$ we define the Lee filter as:

$$lee_r(a_i) = \beta \cdot a_i + (1 - \beta) \cdot avg_r(a_i) \quad (12)$$

So for the new, filtered sequence $(q_{i_{lee}})$ we receive:

$$(q_{i_{lee}}) = (lee_r(a_0), lee_r(a_1), \ldots, lee_r(a_n)) \quad (13)$$

Laplace Filter

A Laplace filter is usually utilized for signal and image processing in order to emphasize edges [VYB1989]. It is based on the *Laplace operator*, which simply means the second derivative in the context of 1D-functions.

Hence 0 as the second derivate points to a local minimum or maximum. This again gives a hint for an edge inside a signal or an image. So the discretization of the second partial derivative results in:

$$\begin{aligned}\Delta f(i) &= \frac{\partial^2 f(i)}{\partial i^2} \\ &\approx \frac{\partial(f(i+1)-f(i))}{\partial i} \\ &\approx f(i+1)-f(i)-(f(i)-f(i-1)) \\ &= f(i+1)-2\cdot f(i)+f(i-1)\end{aligned} \tag{14}$$

Consequently the Laplace operator can be described as a convolution matrix.

$$D_i^2 = \begin{bmatrix}1 & -2 & 1\end{bmatrix} \tag{15}$$

An extension of equation (14) allows determining edges with varying properties.

$$\Delta f_{r,t}(i) = (f(i+r)-2\cdot f(i)+f(i-r))^t \tag{16}$$

Variable $r \in \mathbb{N}$ extends or reduces the radius for the local minimum and maximum search. Parameter $t \in \mathbb{N}$ has a further influence on the filtering process. For instance $t = 2$ leads to only positive results.
Let (a_i) with $a_i \in \mathbb{N}$, $i = 0,\ldots,n$ and $f(i) = a_i$, where $\Delta f_{r,t}(i)$ is undefined for $(i-r) < 0$ or $(i+r) > n$. Now by these preconditions a Laplace filtered sequence $(q_{i_{lpc}})$ based on equation (16) can be determined:

$$(q_{i_{lpc}}) = (\Delta f_{r,t}(0), \Delta f_{r,t}(1), \ldots, \Delta f_{r,t}(n)) \tag{17}$$

6 EXPERIMENTS

This section focuses on accuracy and runtime performance of our system with respect to the filters introduced in section 5.

Motion Filtering and Transformation

Figure 4 shows filtered example 1D-functions on the left and corresponding transforms on the right side. Moreover the basic 1D-function stems from a person's motion while using a wrench. Particularly the charts in figure 4 plot x-axis coordinates of centroids and capture the main motion. It is obvious that the 1D-functions correspond to the left-right and right-left movements. Transforming these 1D-functions by fast Fourier transform (FFT) results in a frequency domain with peaks at 13 and 27. The first amplitude peak at 13 corresponds to the number of left-right movements. In addition the second peak at 27 arises from a slight centroid movement along the x-axis between two repetitions. This typical centroid movement results from the overall body motion.
Without a filter the spatio-temporal motion trajectory has many highs and lows inside a small time frame. If we consider maximum, median or average filter, these highs and lows disappear and the original chart appears smoothed. In addition maximum and medium filter lead on to edged charts. For each filter the corresponding high frequency domain has lower amplitudes than the original high frequency domain without filter usage. Especially the average filter reduces amplitudes of the high frequency ranges. However Lee filter smoothens only parts of the 1D-function, which are below a predefined noise level. Other parts with strong movements even inside small time frames stay nearly unmodified. So only high frequency amplitudes belonging to noisy parts are reduced.
The last chart in figure 4 shows the transform for Laplace filter. Frequency 27 is emphasized strongly, because corresponding edges in the 1D-function are emphasized. By using a small or large radius it is even possible to focus on high frequency or low frequency domains, respectively.

Motion Occlusion

Figure 3 illustrates how occlusion changes motion detection pictures for video scenes. A planing video, with the main motion taking place along the horizontal axis, is occluded by a vertical bar. The occluded motion area is not visible inside the motion detection picture and therefore its image moment and depending 1D-functions change. We adjust the alignment and the width of the bar manually for each class in order to achieve a maximal distraction of the motion centroid. This means the bar has always a relative thickness to the main motion area as shown in figure 3 and furthermore that this bar is always in the middle of the motion.

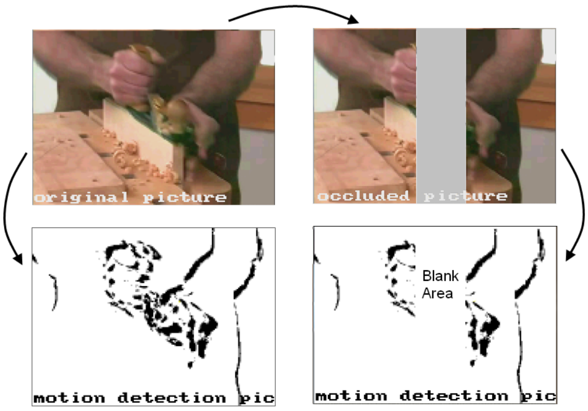


Figure 3: Regions with movement for an occluded planing video

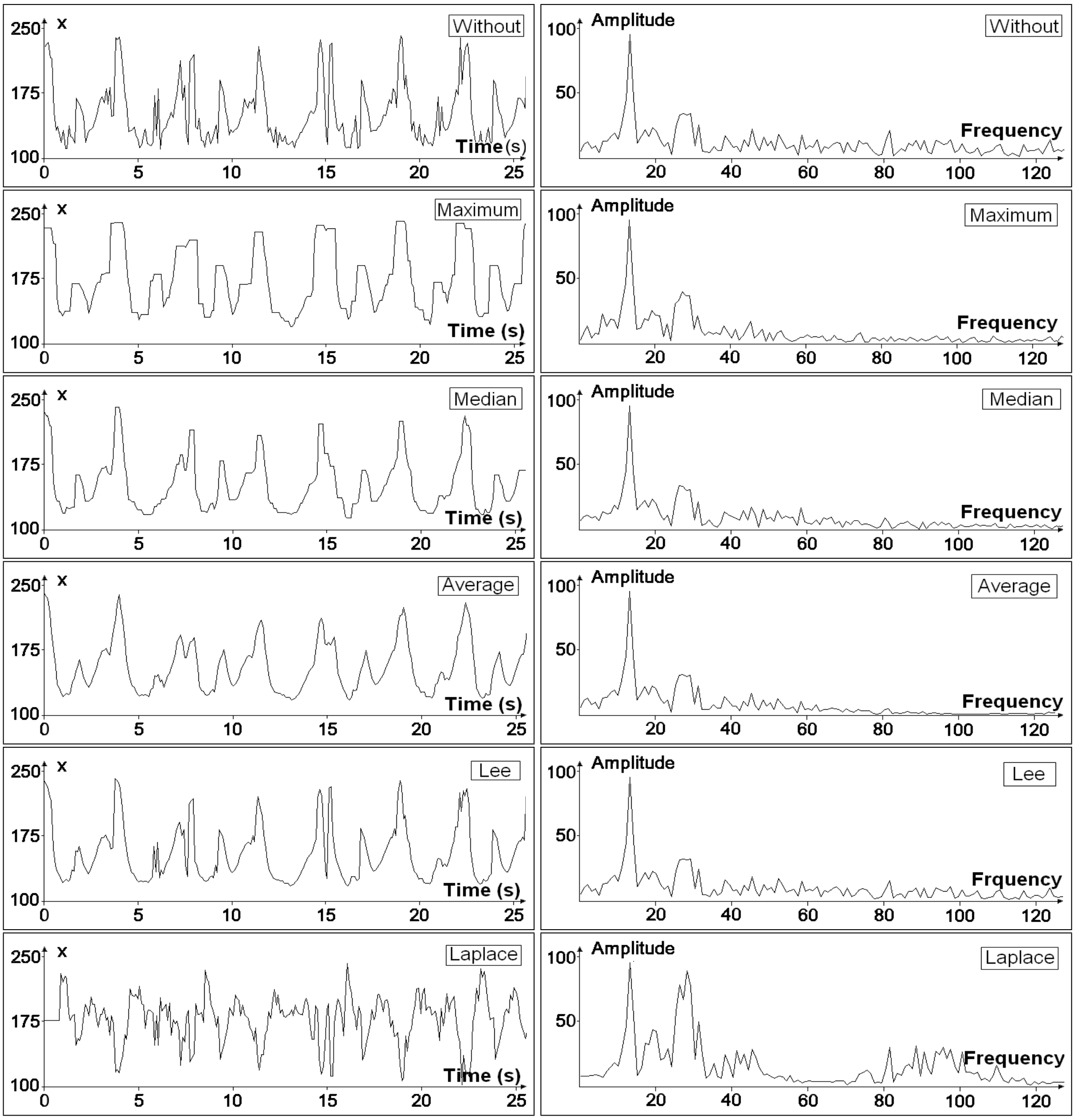


Figure 4: Filtered wrench handling 1D-functions with corresponding transforms

Filter Accuracies

In total we assign 200 own and 102 external videos [YT2010] to one out of ten home improvement classes. These classes contain following activities: filing, hammering, planing, sawing and using a paint roller, paste brush, putty knife, sandpaper, screwdriver, wrench. For our own video data we use twenty-fold cross validation, whereas the external videos are assigned directly to the video classes, because cross validation was not possible due to classes with just too few video clips.

Table 1 shows resulting accuracies for different 1D-functions and filters. Here accuracy means the correct classification ratio. Additionally we check the same video classes with occlusion. Our purpose is to find out, how occlusion affects the classification process and how far filter can balance out irregularities caused by occlusion.

At first glance it becomes apparent that own videos achieve much higher accuracies than external videos. The reason for this behavior is that all own videos have similar recording conditions, whereas all external videos have different recording conditions. Therefore extracted features for external videos vary more than for own videos.

The experimental results in table 1 depict, that occlusion decreases accuracies. But the system is still able to classify own videos via centroid location and

Filter	None	Maximum	Median	Average	Lee	Laplace
Own Videos						
Direction	0.89	0.87	**0.86**	**0.92**	0.89	**0.86**
Location	0.81	**0.72**	0.73	0.73	0.81	**0.84**
Speed	0.48	0.45	**0.37**	**0.49**	0.47	0.44
Own Videos with Occlusion						
Direction	0.70	0.71	0.73	0.75	**0.81**	0.71
Location	0.72	**0.67**	0.69	**0.67**	**0.73**	0.68
Speed	0.35	0.39	0.35	**0.43**	0.36	**0.33**
External Videos						
Direction	0.28	0.22	0.27	**0.20**	0.27	0.26
Location	0.37	0.29	0.36	0.33	**0.39**	**0.26**
Speed	0.23	**0.17**	0.22	0.21	0.23	0.22
External Videos with Occlusion						
Direction	0.25	0.21	0.24	**0.16**	0.18	0.25
Location	0.37	0.32	0.34	0.33	0.37	**0.26**
Speed	0.21	**0.25**	0.21	0.21	**0.25**	**0.18**

Table 1: Overall accuracies for different filter types and 1D-functions

direction based 1D-functions properly. Furthermore for each 1D-function of our own videos there is at least one filter type that increases the accuracy. Especially for occluded videos classified by directional motion data we measure a significant accuracy increase. In this case Lee filter raises the accuracy from 0.70 to 0.81.

For occluded videos and 1D-functions derived by the speed of image moments there is a further significant increase. Here the average filter increases the accuracy from 0.35 to 0.43. With respect to external video data there are only three cases with an accuracy improvement. External videos contain more irregular motions, which again means that for instance the maximum filter substitutes values by maximal noise values and increases therefore the number of false classifications. Moreover the Laplace filter emphasizes noise and the average filter reduces important high frequency amplitudes, which are typical for some external videos.

An overall comparison of all filters leads to the result that the Lee filter is the most accurate filter for repeating motion based video classification. Accuracy increases can be strong and decreases are slight. Here the selective noise reduction seems to be effective. On the other hand Laplace filter tends to increase noise. Hence almost all experimental results show up accuracy decreases. Besides the average filter works only for videos containing clear and smooth motion.

Table 1 shows that Lee filter raises accuracy by 0.11 for directional centroid data of own and occluded videos. Average filter raises accuracy by 0.08 for 1D-functions based on the centroid's speed. By contrast 1D-functions based on the centroid's location do not show any remarkable accuracy raise by applying filters. The reason is that an occlusion influences location based 1D-functions in various ways. Different parts of the frequency domain can be emphasized or declined, whereby filters cannot compensate these changes. Beyond that the location based 1D-functions are the most robust ones, because an occlusion has a minor effect on the overall motion trajectory.

By adding occlusion to video frames the centroid's speed is often raised. This leads to clearer highs and lows inside the 1D-function. Considering that speed information in general is noisy, these clear highs and lows become only apparent in the frequency domain, when the average filter is applied.

Furthermore occlusion weakens the clarity of motion, consequently the centroid direction becomes noisy. Most often this noise stays below a certain amplitude value, so that the Lee filter can remove exactly this specific noise type. This improvement becomes even more apparent, if the original movement without occlusion was wide and clear. In figure 5 classes paint roller, plane and wrench confirm this behavior. Since we consider 10 classes with 20 videos, the maximal number of proper classifications is 20 for each class.

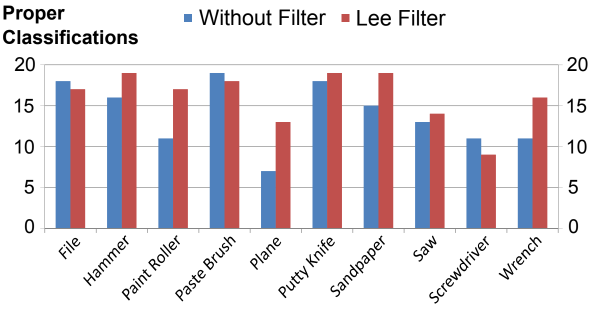


Figure 5: Number of proper classifications with and without Lee filter for occluded own videos

Concluding we can state that filtering 1D-functions can improve accuracy in some cases, but on the whole filters reduce the system's accuracy. They reduce the information content or emphasize noise for motion trajectories, so that the resulting feature vectors cannot be assigned properly.

Runtime Analysis

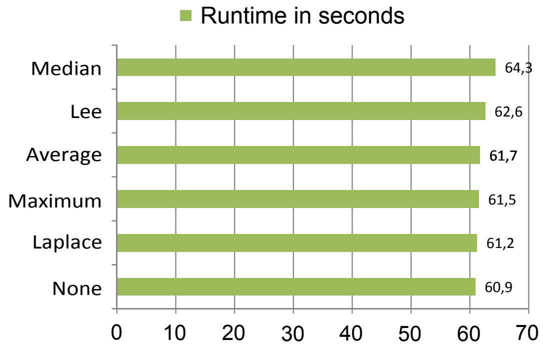


Figure 6: System's runtime with different filters

Figure 6 shows runtime results for each introduced filter. For runtime analysis a 2.2 GHz CPU is used. We assign 1000 videos to one out of 10 classes containing home improvement video data (see figure 5). We reuse our 200 videos covering database five times. Each class consists of 20 videos and each video again consists of 512 frames with a 320 × 240 resolution. Moreover the filter radius is set to 10. Depicted filter runtimes are averages of five separate test iterations. Averaging is necessary, since runtime differences are marginal and system operations can influence the runtime.

Figure 6 shows up small runtime increases, when filters are applied. Standard classification without filter takes 60.9 seconds for 1000 videos. Applying Laplace, maximum or average filter the runtime increase stays below 1 second. These three filters have got similar algorithmic setups. Utilizing Lee filter runtime is 62.6 seconds and therefore longer than the runtime for the previous three filters. Due to additional operations in order to find out the variance, Lee filter requires more runtime. Further on we measure a maximum runtime at 64.3 seconds for median filter. The median filter has to arrange data values in order to find a median. Sorting data values needs more operations than calculating the variance. Thus median filter takes more runtime than Lee filter.

7 CONCLUSION

In this paper we have shown a video classification system based on the frequency of repeating movements. Frequency spectra are computed by transforming spatio-temporal image moment trajectories (1D-functions). The experimental part focused on filtering 1D-functions in order to receive more decisive frequency domains. Test results show that the Lee filter performs best, since this filter smoothens only noisy parts of a 1D-function. However maximum or Laplace filter reduce the system's accuracy in most cases, because either high frequencies are smoothed too strongly or noisy parts are emphasized, respectively. Runtime analysis turns out that Lee filter needs more operations than maximum, average or Laplace filter, but less operations than median filter. Applying filters to 1D-functions can improve the system's accuracy in some cases, but in general the accuracy is decreased. Particularly smoothing filters like maximum, median and average filter reduce the information content.

But there are still edge detection filters as the Prewitt filter or noise removing filters as the harmonic mean filter, which have to be analyzed and could reveal more accurate test results.

8 REFERENCES

[AS2001] Alsultanny, Y. and Shilbayeh, N., Examining filtration performance on remotely sensing satellite images, SSIP, pages 75–80, 2001.

[AC2012] Ayyildiz, K. and Conrad, S., Video classification by partitioned frequency spectra of repeating movements, WASET, pages 154–159, 2012.

[CCK2004] Cheng, F., Christmas, W., and Kittler, J., Periodic human motion description for sports video databases, ICPR, pages 870–873, 2004.

[FZP2005] Fanti, C., Zelnik-Manor, L., and Perona, P., Hybrid models for human motion recognition, CVPR, pages 1166–1173, 2005.

[LEE1980] Lee, J., Digital image enhancement and noise filtering by use of local statistics, TPAMI, pages 165–168, 1980.

[MAS1985] Mastin, G. ,Adaptive filters for digital image noise smoothing: An evaluation, CVGIP, pages 103–121, 1985.

[MLH2006] Meng, Q., Li, B., and Holstein, H., Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach, IVC, pages 795–809, 2006.

[VUE2010] Vanithamani, R., Umamaheswari, G., and Ezhilarasi, M., Modified hybrid median filter for effective speckle reduction in ultrasound images, ICNVS, pages 166–171, 2010.

[VYB1989] Vliet, L., Young, I., and Beckers, G., A nonlinear Laplace operator as edge detector in noisy images, CVGIP, pages 167–195, 1989.

[WSL1995] Wong, W., Siu, W., and Lam, K., Generation of moment invariants and their uses for character recognition, PRL, pages 115–123, 1995.

[YT2010] YouTube, L., Youtube: Broadcast yourself, `www.youtube.com`, 2010.

Efficient Linear Local Features of Digital Signals and Images: Computational and Qualitative Properties

Vladislav Myasnikov
Samara State Aerospace University
Moskovskoye shosse 34
Russia 443086, Samara
vmyas@geosamara.ru

ABSTRACT

The paper presents the analysis of efficiency of two original approaches to the construction of the sets of linear local features (LLF), which are used for digital signal and image processing. The first approach is based on generating of LLF set, which consists of separately constructed efficient LLFs, each of which has its own algorithm for feature calculation. The second approach assumes the construction of an efficient LLF set, which has a single algorithm for joint simultaneous computation of all features. The analysis is carried out by several indicators that characterize the computational and qualitative properties of the constructed LLFs.

Keywords

Features, digital images and signals, computational complexity, processing quality.

1. INTRODUCTION

Feature creation is one of the main stages of visual data processing systems development and it affects the final quality of the system. A local feature of a digital signal is usually a numerical characteristic - the result of a transformation of digital signal/image samples, which belong to a local analysis area [1]. For linear local features (LLF) this transformation is linear with constant parameters. Taking into account, that calculation of LLF values can be made in different ways (direct algorithms or fast convolution, recursive algorithms, etc.), a specific LLF is characterized by two components – a linear convolution kernel (we call it as *LLF's kernel*) and an algorithm for calculation of the convolution of the input signal/image and this kernel (we call it as *LLF's algorithm* or *algorithm for LLF values calculation*). Moreover, if LLF's kernel determines *qualitative characteristics* of the specific LLF, the algorithm for LLF values calculation characterizes *computational complexity* of the feature. Sets of features, which have not just one but several feature values for the same analysis area of a digital signal, are usually used to solve practical problems. It is essential, that calculation of the corresponding feature values in a set can be produced by several independent algorithms as well as a general algorithm that executes jointly simultaneously calculations for all the values of features in a set. In the latter case we speak about a *set of jointly computed features*. Qualitative indicators (for sets of jointly and independently calculated LLFs) are determined by a set of corresponding kernels. The general formulation of the *problem of constructing an efficient (set of) LLFs* implies the constructing LLFs (or set of LLFs) with the best quality indicator and with specified computational complexity [2-4]. Despite the seeming simplicity of the presented formulation, we should accept the problem of constructing features and their sets extremely complex.

In the author's paper [2] the formal approach for efficient LLFs construction has been proposed, and in the papers [3, 4] this approach has been extended to the case of constructing an efficient set of jointly calculated LLFs. These approaches allow us to design an efficient LLF (or efficient set of LLFs) for the most applied problems. The term "*efficiency of LLF*" refers to the satisfaction of *two basic requirements*:

- algorithm for LLF values calculation has a predetermined computational complexity value;
- LLF's kernel(s) is(/are) the best matched to a given quality indicator.

Under the preceding requirements efficient LLFs enable us to establish a reasonable balance between two opposing groups of features:

- features, which are optimal in the sense of some quality criteria and do not have suitable or fast computation algorithm (e.g., features, obtained using Karhunen-Loeve transform);
- features, which are obtained by using fast algorithms and are not related to the content of the

problem and relevant quality indicators (e.g., features, obtained using fast Fourier transform algorithm).

According to the information of author, the only alternative approach of the feature construction, that satisfies all requirements mentioned above, exists. It was proposed by Prof. V.Labunets in 2013 and was denoted as «multiparametric wavelet transforms» [12,13]. Unfortunately, these papers do not provide the method of solving the efficient LLFs construction problem, they only show that multiparametric (or adaptive) wavelets exist and can be constructed.

The *main purpose of this paper* is to analyze/compare the author's two approaches to constructing sets of LLFs. The first approach constructs a set of features by constructing a set of efficient LLFs, each of which has its own algorithm for feature calculation. The second approach constructs an efficient set of LLFs, in which there is a single algorithm for computing all features jointly. Short description of these approaches is presented in the Section 2, where the known information is collected. New results on analytical and experimental analysis of these approaches are presented in Sections 3 and 4.

2. SETS OF JOINTLY AND INDEPENDENTLY CALCULATED LINEAR LOCAL FEATURES OF DIGITAL SIGNALS: BACKGROUND

This Section presents short reference information on the efficient linear local features of the digital signals: basic definitions, equations and construction methods. Full description may be found in the papers [2-4].

Let **N** be a set of natural numbers, **K** be a commutative ring with unity, $\{x(n)\}_{n=0}^{N-1}$ be an input signal of length *N* over the ring **K**.

Definition 1. A *linear local feature (LLF) of length M over the ring* **K** is a pair $\left(\{h(m)\}_{m=0}^{M-1}, A\right)$, where $\{h(m)\}_{m=0}^{M-1}$ is a linear convolution kernel of length *M*, which is determined as a finite sequence over the ring **K** and satisfies the constraint $h(m) \neq 0, h(M-1) \neq 0$, and *A* is an algorithm for calculating a linear convolution (1) of an arbitrary input signal over the ring **K** with the kernel $\{h(m)\}_{m=0}^{M-1}$:

$$y(n) = \sum_{m=0}^{M-1} h(m) x(n-m), \quad n = \overline{M-1, N-1}. \tag{1}$$

A *set of R independently calculated LLF* of length M over the ring **K** is a further set of LLFs:

$$\left\{\left(\{h_r^{ind}(m)\}_{m=0}^{M-1}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}.$$

Definition 2. A *set of R jointly calculated LLFs over the ring* **K** is a pair $\left(\{h_r(m)\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}}, A\right)$, where $\{h_r(m)\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}}$ is a set of *R* kernels, each of which is determined as a finite sequence over the ring **K** and satisfies the following constraints:

$$h_0(0) \neq 0; \quad \forall r \in \overline{0, R-1} \quad \exists m \in \overline{0, M-1} \quad h_r(m) \neq 0;$$
$$\exists\, r \in \overline{0, R-1} \quad h_r(M-1) \neq 0;$$

and *A* is an algorithm for joint calculation of a set of linear convolutions of an arbitrary input signal $\{x(n)\}_{n=\overline{0,N-1}}$ $(M < N)$ over the ring **K** with a set of kernels:

$$y_r(n) = h_r(n) * x(n) = \sum_{m=0}^{M-1} h_r(m) x(n-m), \qquad n = \overline{M-1, N-1}, \quad r = \overline{0, R-1}. \tag{2}$$

To distinguish the elements of sets of independently calculated LLFs from jointly calculated LLFs the last will be denoted as follows:

$$\left(\{h_r^{set}(m)\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}}, A^{set}\right) \square$$

In author's papers [2-4] we proposed a method for construction of the sets of independently and jointly calculated LLFs, based on designing (sets of) sequences of kernel's samples in the form of *linear (mutual) recurrent sequences* (LRS or LMRS, respectively) [5,6,9]. For these (sets of) sequences, called NMC-(sets) sequences[1], the computational complexity of calculating linear convolutions (1) or (2) is minimal. For fixed parameters of *linear (mutual) recurrent relations* (LRR or LMRR, respectively) these sets of NMC sequences or NMC-sets of sequences form a collection of sequences, denoted, respectively $\wp(M, K, \bar{c})$ or $\wp(R, M, T, K, \bar{a})$. Here *K* is an order of LRR for samples of a sequence, *R* is a number of sequences in a set, *T* is an order of mutual recurrence (for sets), $\bar{c}$ and $\bar{a}$ are LRR's or LMRR's coefficients respectively. As it has been shown in papers [2-4], the powers of these collections satisfy the relations:

$$\forall M > K \geq 1, \bar{a}(a_K \neq 0) \quad \left|\wp(K, M, \bar{c})\right| \leq C_{M+K-2}^{K-1},$$
$$\forall M > K \geq 1 \quad R \geq T \geq 1$$
$$\left|\wp_{(b,c,d)}(R, M, T, K, \bar{a})\right| \leq C_{R(M+K)-1}^{RK} - C_{R(M+K-1)-1}^{RK}. \tag{3}$$

[1] NMC - normalized with minimal complexity

Each sequence from the collection, along with its parameters, is also characterized by a set Θ of additional independent parameters – *degrees of freedom*. The powers of degrees of freedom sets Θ are determined in the following way [2-4]:

$$\left|\Theta_{\wp(M,K,\overline{c})}\right| = K, \quad \left|\Theta_{\wp(R,M,T,K,\overline{a})}\right| = RK \,. \tag{4}$$

The computational complexity of algorithms A^{ind} and A^{set} for calculating relevant features or sets of features for all NMC sequences or NMC sets of sequences from collections $\wp(M,K,\overline{c})$ and $\wp(R,M,T,K,\overline{a})$ is determined by these equations [2-4]:

$$u\left(A^{ind}\right) \le 2K \frac{N}{N-M+1}, \tag{5}$$

$$u\left(A^{set}\right) \le \frac{N}{N-M+1} \left(\begin{array}{l} R(K-1) - (R-1)\xi_{add} + \\ + (K+1)T\left(R - \dfrac{T-1}{2}\right) \end{array} \right). \tag{6}$$

The *problems* of construction of an *efficient (set of) LLF(s)* are defined as follows [2-4]. A *particular problem of construction of an efficient set of LLFs* is defined as a problem of searching in a predefined collection $\wp(R,M,T,K,\overline{a})$ of such a set (with its corresponding algorithm of joint calculation of LLFs A^{set}), for which the minimum condition for a problem-specific objective function $\Psi : \mathbf{K}^{RM} \to \mathbf{R}$ is fulfilled:

$$\Psi(h_0(0),\ldots,h_0(M-1),\ldots,h_{R-1}(0),\ldots,h_{R-1}(M-1)) \\ \to \min_{\left\{h_r^{set}(m)\right\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}} \in \wp(R,M,T,K,\overline{a})} . \tag{7}$$

For *a particular problem of construction of an efficient LLF* the drafting changes are related to a collection $\wp(M,K,\overline{c})$ and an objective function $\Psi : \mathbf{K}^{M} \to \mathbf{R}$.

The difference in the solutions of these problems lies in the fact that in the first case a set of jointly calculated LLFs is formed $\left(\left\{h_r^{set}(m)\right\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}}, A^{set}\right)$ and in the second case there is only one LLF constructed $\left(\left\{h(m)\right\}_{m=0}^{M-1}, A\right)$. Note that using a particular problem of constructing an efficient LLF it is possible to construct a set of independently calculated features $\left\{\left(\left\{h_r^{ind}(m)\right\}_{m=0}^{M-1}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}$, for example by their consequent construction with appropriate modification of objective functions for each of particular problems.

The computational complexity of calculation of the sets of LLFs and the number of their degrees of freedom can be used as indicators or constraints in the analysis of constructed sets of jointly and independently calculated LLFs. Additionally, for further analysis we can introduce a formalized notion of collections "comparability" of jointly and independently calculated LLFs as follows.

Let's consider a set of LMRS $\left\{h_r(m)\right\}_{m=\overline{0,M-1}}$ $\left(r=\overline{0,R-1}\right)$, which belongs to collection $\wp(R,M,T,K,\overline{a})$ and satisfies a LMRR [3,4]:

$$h_r(m) = \begin{cases} b_{rm}, \quad r=\overline{0,T-1},\ m=\overline{0,K-1}, \\ \displaystyle\sum_{k=1}^{\min(K,m)} a_{0k}^r h_r(m-k) + \\ \displaystyle + \sum_{t=1}^{\min(r,T-1)} \sum_{k=0}^{\min(K,m)} a_{tk}^r h_{r-t}(m-k) + \varphi_r(m), \\ \qquad\qquad r \ge T \vee m \ge K. \end{cases} \tag{8}$$

In case, when $\varphi_r(m) \equiv 0$, LMRS and LMRR are called *homogeneous* [5,6,9]. The following lemma defines characteristics of the sequences in this set.

Lemma (on solution of homogeneous LMRR). Let *T*=*R*≥1 and a homogeneous LMRR of order (*T*,*K*)

$$h_r(m) = \sum_{k=1}^{K} a_{0k}^r h_r(m-k) + \sum_{t=1}^{r} \sum_{k=0}^{K} a_{tk}^r h_{r-t}(m-k), \\ r=\overline{0,R-1}$$

determines the samples of the collection of *R* sequences $\left\{h_r(m)\right\}_{\substack{r=\overline{0,R-1};\\ m=0,1,\ldots}}$ for the entire domain. Let us define matrixes $Q_r(z)$ of size $r \times r$, where each element $q_{ij}^r(z)$ is determined $\left(q_{ij}^r(z) \equiv q_{ij}^t(z) \quad \forall i,j < \min(r,t)\right)$ with an expression:

$$q_{ij}^r(z) = \begin{cases} \displaystyle\sum_{k=1}^{K} a_{0k}^i z^{-k} - 1, & i=j, \\ 0, & i<j, \quad i,j=\overline{0,r-1}. \\ \displaystyle\sum_{k=0}^{K} a_{(i-j)k}^i z^{-k}, & i>j, \end{cases}$$

Then every r-th sequence of the collection for the entire domain satisfies the following homogeneous LRR:

$$h_r(m) = \sum_{s=1}^{K(r+1)} c_s^{r+1} h_r(m-s), \quad r=\overline{0,R-1},$$

where the values $\left\{c_s^r\right\}_{s=1}^{Kr}$ are coefficients in the matrix $Q_r(z)$ determinant:

$$\det(Q_r(z)) = \prod_{i=0}^{r-1}\left(\sum_{k=1}^{K} a_{0k}^i z^{-k} - 1\right) = 1 - \sum_{s=1}^{Kr} c_s^r z^{-s} \,. \qquad ■$$

It is obvious, that under the lemma’s conditions, the sequence of the collection with number *r* satisfies the homogeneous LRR with order not exceeding $K(r+1)$. This proved connection allows us to give the following definition for “*comparability*” of jointly and independently calculated LLF collections.

Definition 3. A set of collections of LRSs $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,R-1}}$ and a collection of LMRRs $\wp\left(R, M, T, K, \bar{a}\right)$ are called *comparable*, if these equations are valid:

$$K_r = K(r+1),$$

$$\prod_{i=0}^{r-1}\left(\sum_{k=1}^{K} a_{0k}^{i} z^{-k} - 1\right) = 1 - \sum_{s=1}^{Kr} c_s^r z^{-s}, \quad r = \overline{0, R-1}.$$

The fact of compatibility means that one can specify for at least one (homogeneous) set of sequences from $\wp\left(R, M, T, K, \bar{a}\right)$ exactly the same set of sequences from $\left\{\wp\left(M, K_r, \bar{c}_r\right)\right\}_{r=\overline{0,R-1}}$. Note also that although there are more than one equal sets of sequences for comparable collections the full match of sets of sequences doesn’t happen.

The results of this section allow us to make an analytical comparison of comparable sets of collections.

3. COMPUTATIONAL AND QUALITATIVE PROPERTIES: ANALYTICAL COMPARISON

3.1 Comparison of Linear Local Features Sets for Comparable Collections

Let $N, R, M, T, K \in \mathbf{N}$, and $\left(\left\{h_r^{set}(m)\right\}_{\substack{r=\overline{0,R-1};\\ m=\overline{0,M-1}}}, A^{set}\right)$ is an arbitrary efficient set of LLFs for a collection $\wp\left(R, M, T, K, \bar{a}\right)$. Computational complexity of the algorithm of calculation of the LLF, corresponding to any set of sequences of this collection, satisfies the equation (6). From the other hand, one can construct independent efficient LLFs $\left\{\left(\left\{h_r^{ind}(m)\right\}_{m=\overline{0,M-1}}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}$ from the comparable $\wp\left(R, M, T, K, \bar{a}\right)$ set of collections $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,R-1}}$. Then, taking into account equations (5), computational complexity of LLF set calculation $\left\{\left(\left\{h_r^{ind}(m)\right\}_{m=\overline{0,M-1}}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}$ is determined as follows:

$$\sum_{r=0}^{R-1} u\left(A_r^{ind}\right) \le \frac{N}{N-M+1} KR(R+1). \tag{9}$$

Comparing the right part of this equation with the equation (6), one can assure of the following relation correctness:

$$\begin{pmatrix} R(K-1) - (R-1)\xi_{add} + \\ \qquad + (K+1)T\left(R - \dfrac{T-1}{2}\right) \end{pmatrix} < KR(R+1). \tag{10}$$

Then the following statement is correct.

Statement 1. Let $K, R, M, T \in \mathbf{N}$, $K \ge 1$, $R \ge T \ge 2$, sets of LLFs $\left(\left\{h_r^{set}(m)\right\}_{\substack{m=\overline{0,M-1},\\ r=\overline{0,R-1}}}, A^{set}\right)$ and $\left\{\left(\left\{h_r^{ind}(m)\right\}_{m=0}^{M-1}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}$ are constructed for comparable collections $\wp\left(R, M, T, K, \bar{a}\right)$ and $\left\{\wp\left(M, K_r, \bar{c}_r\right)\right\}_{r=\overline{0,R-1}}$ correspondingly, while relations (5) and (6) are satisfied as equalities. Then

$$u\left(A^{set}\right) < \sum_{r=0}^{R-1} u\left(A_r^{ind}\right). \tag{11}$$

This statement makes it possible to confirm the potential *computationally* benefits of jointly calculated LLFs in comparison with sets of independently calculated efficient LLFs designed for comparable collections.

3.2 Comparison of Linear Local Features Sets with Equal Number of Degrees of Freedom

Equation (4) means that the number of degrees of freedom for the specific efficient set of LLFs from the collection $\wp\left(R, M, T, K, \bar{a}\right)$ is equal to *KR*. From the other hand, one can construct $\tilde{R}$ independent efficient LLFs from collections $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,\tilde{R}-1}}$ in such a way, that the overall number of degrees of freedom becomes equal *KR* too. It is easy to prove that in this case the following equality is valid:

$$\tilde{R}\left(\tilde{R}+1\right) = 2R. \tag{12}$$

Using (12) one can assure the following relation correctness ($K, R, \tilde{R}, M, T \in \mathbf{N}$, $K \ge 1$, $R \ge T \ge 2$):

$$\begin{pmatrix} R(K-1) - (R-1)\xi_{add} + \\ \qquad + (K+1)T\left(R - \dfrac{T-1}{2}\right) \end{pmatrix} > K\tilde{R}\left(\tilde{R}+1\right).$$

Statement 2. Let $K, R, M, T \in \mathbf{N}$, $K \ge 1$, $R \ge T \ge 2$, jointly and independently calculated LLFs have equal number of degrees of freedom (i.e. equation (12) is correct), while relations (5) and (6) are satisfied as equalities. Then

$$u\left(A^{set}\right) > \sum_{r=0}^{\tilde{R}-1} u\left(A_r^{ind}\right). \qquad (13)$$

This statement makes it possible to confirm the potential *computational* benefits of set of independently calculated LLFs in comparison with the set of jointly calculated efficient LLFs designed for equal number of degrees of freedom.

3.3 Comparison of the Computational Complexity of Solving the Particular Problem of Features Construction

Let $\wp(R, M, T, K, \bar{a})$ and $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,R-1}}$ are comparable collections of jointly and independently calculated LLFs. To compare the calculational complexities of the solving of the particular tasks of LLFs $\left(\left\{h_r^{set}(m)\right\}_{\substack{r=\overline{0,R-1};\\ m=\overline{0,M-1}}}, A^{set}\right)$ and $\left\{\left(\left\{h_r^{ind}(m)\right\}_{m=\overline{0,M-1}}, A_r^{ind}\right)\right\}_{r=\overline{0,R-1}}$ construction (see Section 2), we have to compare the number of sequences in the collections $\wp(R, M, T, K, \bar{a})$ and $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,R-1}}$. In the case of the collection $\wp(R, M, T, K, \bar{a})$ the number of sequences is defined by equation (3). When we form the set of sequences from the collections $\left\{\wp\left(M, K_r, \bar{c}^{\,r}\right)\right\}_{r=\overline{0,R-1}}$, we can use two obvious strategies:

- *exhaustive search* (optimal solution)*:* in this case the number of sequences sets takes the form:

$$\prod_{r=0}^{R-1} \left|\wp\left(M, K(r+1), \bar{c}^{\,r}\right)\right|;$$

- *incremental search* (quasi-optimal solution)*:* in this case we search for the sequence of the *r*-th collection when the sequence of the (*r*-1)-th collection is found. The number of possible sets of sequences has the form: $\sum_{r=0}^{R-1} \left|\wp\left(M, K(r+1), \bar{c}^{\,r}\right)\right|$.

Taking into account equations (3), we can compare the computational complexity of solving the particular problem of LLFs construction by comparing the value $C_{R(M+K)-1}^{RK} - C_{R(M+K-1)-1}^{RK}$ with $\prod_{r=0}^{R-1} C_{M-2+(r+1)K}^{(r+1)K-1}$ (exhaustive search case) or $\sum_{r=0}^{R-1} C_{M-2+(r+1)K}^{(r+1)K-1}$ (incremental search case).

It may be done by analyzing the following ratios:

exhaustive search: $$\frac{C_{R(M+K)-1}^{RK} - C_{R(M+K-1)-1}^{RK}}{\prod_{r=0}^{R-1} C_{M-2+(r+1)K}^{(r+1)K-1}}, \qquad (14)$$

incremental search: $$\frac{C_{R(M+K)-1}^{RK} - C_{R(M+K-1)-1}^{RK}}{\sum_{r=1}^{R} C_{M-2+(r+1)K}^{(r+1)K-1}}. \qquad (15)$$

Using (15) we can prove the following statement.

Statement 3. Let $K, R, M, T \in \mathbf{N}$ $K \geq 1,\quad R \geq T \geq 2,\quad M > RK + 1$. Then

$$C_{R(M+K)-1}^{RK} - C_{R(M+K-1)-1}^{RK} > \sum_{r=0}^{R-1} C_{M-2+(r+1)K}^{(r+1)K-1}.$$

This statement makes it possible to confirm that solving of the particular problem of jointly calculated LLFs construction is more difficult than the solving of the particular problem of independent calculated LLFs. Direct numerical analysis of the ratio (15) for useful parameters range (*M*=21...32; *R*=1..4) shows that it is much more difficult: values of the ratio (15) are in the range [1, 5.7*10^9].

Unlike the situation is considered with an incremental search, it the case of exhaustive search it is not possible to make an unambiguous conclusion. Direct numerical analysis of the ratio (14) for parameters ranges mentioned above shows that it is in the range [7.2*10^-8, 3.97].

Finally, we can conclude that:

- *quasi-optimal solution* of the particular problem of independently calculated LLFs construction, based on the incremental search, is less difficult then the optimal solution of the particular problem of jointly calculated LLFs construction;
- *optimal solution* of the particular problem of independently calculated LLFs construction, based on the exhaustive search, may be radically difficult then the optimal solution of the particular problem of jointly calculated LLFs construction. So, when we are going to find optimal solution, jointly calculated LLFs are preferable.

3.4 Analytical Comparison: Conclusion

Analytical and numerical results presented in this Section above make it possible to conclude that the analytical analysis cannot provide the unambiguous answer on the question what type of LLFs (sets of independently or jointly calculated LLFs) is better. Therefore, we are trying to answer this question using experiments.

4. COMPUTATIONAL AND QUALITATIVE PROPERTIES: EXPERIMENTAL COMPARISON

In order to complete the comparison of the sets of independently and jointly calculated LLFs and to compare them with existent typical ways of linear local features calculations we will consider several illustrative tasks. In every task we will compare computational and qualitative properties of the constructed LLFs.

Despite of the illustrative character of the chosen tasks, they appear often in real applications in similar formulations, and explicit criteria and mathematical model of the processing signal is necessary only to point out the best (from typical ways of linear local features calculations) set of feature kernels.

So, general problem statement is as follows. Let we have a digital signal that may be interpreted as a realization of the discrete stationary random process $X(n)$ with zero mean and autocorrelation function:

$$R(n) = D_x \rho^{|n|}, \quad n \geq 0\,, \tag{16}$$

here $D_x = 1, \quad \rho = 0{,}95$, for definiteness. We allow that the length of the processing signal N is unlimited and to perform the local analysis of the signal in the specific position n_0 we have to use M=33 samples of the signal (i.e. «processing window»): $X(n_0),\ldots,X(n_0+M-1)$. Also, we allow that the quality of the local analysis of the signal depends directly on the *quality indicator*, that is given by the following equation:

$$J_\alpha = \alpha \cdot \frac{E\left(\sum_{m=0}^{M-1}\left(\sum_{r=0}^{R-1} Y_r h_r(m) - X(m)\right)^2\right)}{E\left(\sum_{m=0}^{M-1}(X(m))^2\right)} + \tag{17}$$

$$+(1-\alpha)\frac{2}{R(R-1)}\sum_{r=0}^{R-2}\sum_{t=r+1}^{R-1}\frac{\langle h_t, h_r\rangle^2}{\|h_t\|\cdot\|h_r\|}, \quad (\alpha \in [0,1]).$$

Here $\{h_r(m)\}_{r=\overline{0,R-1};\; m=\overline{0,M-1}}$ is a set of kernels that is used for linear representation of the analyzed fragment of the signal, $E(\ldots)$ - the mathematical expectation operator. Obviously, the less the quality indicator the better the set of features.

In the equation (17) the first term defines relative error of the representation of the signal fragment using weighted sum of LLF's kernels, the second term shows the correlation rate of the kernels, and the denominator of the first term satisfies the equality:

$$E\left(\sum_{m=0}^{M-1} X^2(m)\right) = D_x M \quad (= 33).$$

Let define the *general problem as follows*: we have to obtain the set of kernels $\{h_r(m)\}_{r=\overline{0,R-1};\; m=\overline{0,M-1}}$ and algorithm(s) of calculation of the set of convolutions (2) of the signal with these kernels, which provide minimal value of the quality indicator (17) and satisfy certain restriction on the computational complexity of convolutions (2) calculation:

$$\begin{cases} J_\alpha \to \min \\ u(\ldots) \leq u_{\max}. \end{cases} \tag{18}$$

Bellow, we provide several ways to solve the problem (18). First and second methods (solutions, that are ordinary used in digital signal and image processing) use "optimal" kernels, that comes from Karhunen-Loewe decomposition [7] of the fragment of the discrete stationary random process (16). The only difference between these methods is the convolution algorithms. First method (*method 1*) uses the direct convolution algorithm, and the second one (*method 2*) uses the fast convolution algorithm, that is based on the Fast Fourier Transform (FFT) [8,10] and optimal sectioning of the processing signal [10]. In practice, the second method is the de facto standard for solutions of this type of problems. *Method 3* uses the set of jointly calculated LLF's, and *methods 4-7* use the sets of independently calculated LLF's (description of these methods is given bellow). It should be noted that the detail description of the problem (18) when α=1 using the set of jointly calculated LLF's was given in the paper [4]. Some useful equations, that are used here for calculation of an error of representation of the fragment of the discrete stationary random process using non-orthogonal kernels, were given in that paper too.

We analyze solutions of the problem (18) for three values of parameter α, namely:

- *group 1*: α=1,
- *group 2*: α=0,
- *group 3*: α=1/2.

Solution of the problem (18) using sets of independently or jointly calculated LLFs (methods 3-7) is performed by solving the particular problem (7) of constructing an efficient set of LLFs. This particular problem [2-4] means that the LLF's kernels are from the specific collection, and this collection is defined both by the task restrictions (the size M of the "processing window" and the upper bound $u_{\max}$ of the calculational complexity of features calculation), and subjective chosen parameters $T, K, \overline{a}$ and $\{\overline{c}^r\}_{r=\overline{0,R-1}}$. In our experiments, parameters are as follows:

- *method 3*: collection $\wp(R, M, T, K, \overline{a})$, parameters:

$T = 2, K = 1, \quad a_{01} = 1, a_{10} = 1, a_{11} = a_{10} = 1$;

- *methods 4-7*: collections $\{\wp(M, K_r, \overline{c}^r)\}_{r=\overline{0,R-1}}$, parameters:

- quasi-polynomial (*method 4*):

$c_k^r = (-1)^{k+1} C_{(r+1)K}^k, \quad \left(K = 1, \quad k = \overline{1,(r+1)K}\right)$;

- quasi-exponential (*method 5*):

$c_k^r = ((r+1)K)^{-1} \rho^k, \quad \left(K = 1, \quad k = \overline{1,(r+1)K}\right)$;

- quasi- Fibonacci (*method 6*):

$R=1{,}2:\quad c_1^1 = \ c_1^2 = c_1^1 = 1;$

$R=3:\quad c_1^3 = 1/2,\quad c_2^3 = 3/2,\quad c_3^3 = 1/2;$

$R=4:\quad \ldots$

- quasi-harmonic (*method 7*):

$R=1:\ c_1^1 = 1,$

$R=2:\ c_2^1 = 2\cos(\omega),\quad c_2^2 = -1;$

$R=3:\ c_3^1 = \cos(\omega), c_3^2 = 2\cos^2(\omega)-1, c_3^3 = -\cos(\omega);$

$R=4:\quad \ldots$

Presented collection names are derived from the names of the sequences ($R \ge 2$), that satisfy the homogeneous LMRS (8) with the same parameters.

The calculational complexity of the independently and jointly calculated LLFs is defined by equations (5)-(6), that were used as equalities.

Figures 1,3-5 present the obtained results, that show the dependence of the quality indicator J_α of the constructed features on the computational complexity of the features calculation $u(\ldots)$. These results lead to the following conclusions.

– For the first group of the tasks (α=1, Fig.1) quality indicators for the sets of independently and jointly calculated LLFs (methods 3-4) are significantly less (i.e. the quality is significantly higher) then the quality indicators obtained for «optimal» kernels (obtained using Karhunen-Loewe decomposition) and direct (method 1) or fast (method 2) convolution algorithms. Particularly, when the calculational complexity of the features calculation satisfies $u_{\max} = 40$ the *quality* of the set of jointly calculated LLFs *is six time higher* (vs method 2)! For this particular case, Fig.2 shows four constructed kernels for the jointly calculated LLFs. It is easy to see that these kernels are similar to the «optimal» kernels (sinusoids of different phases and frequencies), that may be obtained using Karhunen-Loewe transform.

– For the first group of the tasks (α=1, Fig.3) quality indicators for the set of jointly calculated LLFs is less (i.e. the quality is higher) then the quality indicators for the sets of independently calculated LLFs.

– For the 2nd and 3rd groups of the tasks (α<1, Figs.4-5) quality indicator for all types of LLFs depends significantly on the collection parameters. Therefore, changing these parameters we can obtain different answers which type of feature sets (set of jointly or set of independently calculated LLFs) is better. In practice, the best type of LLFs may be found using global optimization methods: genetic algorithms, simulated annealing, etc.

The obtained experimental results allow us to make two conclusions:

- the proposed efficient LLFs have advantage in comparison with the traditional way of solving such a type of problems, even when the “optimal” kernels/bases exist;
- jointly and independently calculated efficient LLFs have comparable efficiency, i.e. neither of two approaches has clear advantages.

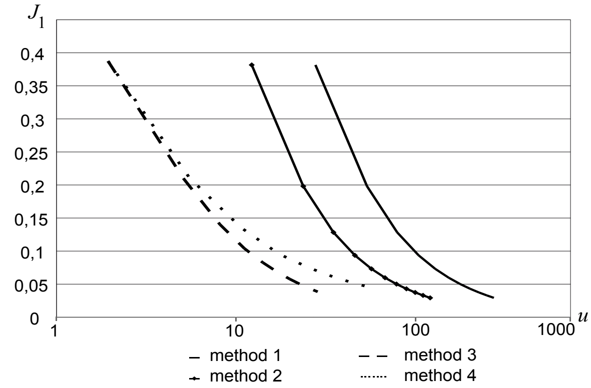


Figure 1. Comparison of the proposed efficient LLFs (methods 3-4) with traditional way of features construction (methods 1-2); task group 1: α=1.

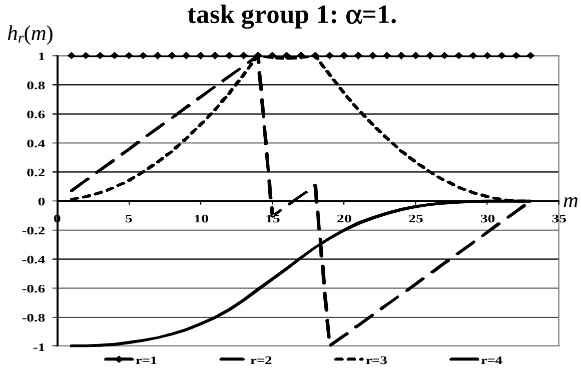


Figure 2. First four constructed kernels for the jointly calculated LLFs (for convenience, we put kernels to the range [-1,1]).

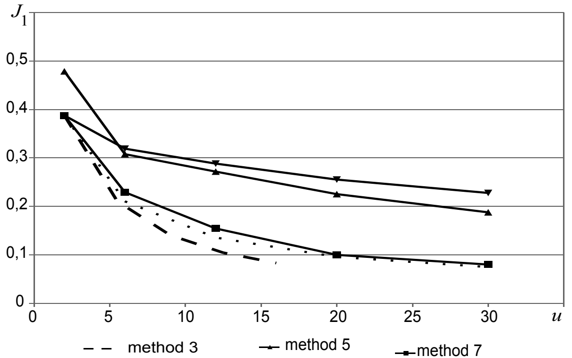


Figure 3. Analysis of the proposed efficient LLFs: comparison of the sets of jointly (method 3) and independently (methods 4-7) calculated LLFs; task group 1: α=1.

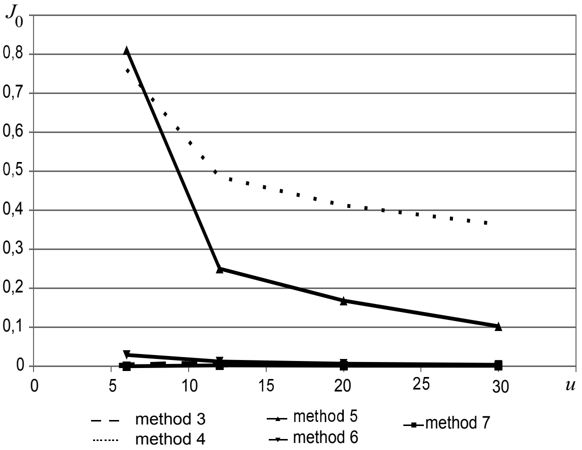


Figure 4. Analysis of the proposed efficient LLFs: comparison of the sets of jointly (method 3) and independently (methods 4-7) calculated LLFs; task group 2: α=0.

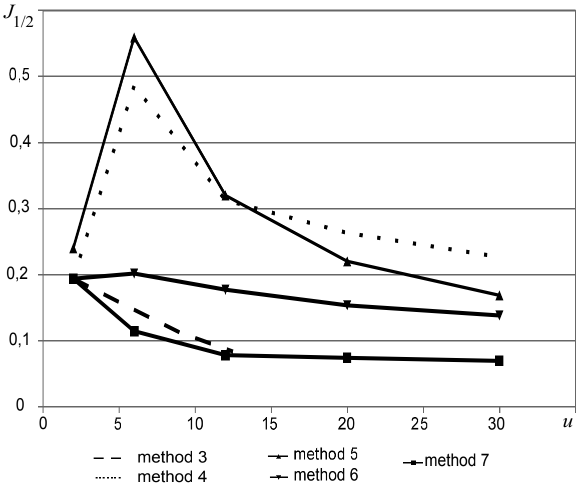


Figure 5. Analysis of the proposed efficient LLFs: comparison of the sets of jointly (method 3) and independently (methods 4-7) calculated LLFs; task group 3: α=1/2.

5. CONCLUSIONS

In this paper two approaches to the construction of a set of linear local features for digital signals are analyzed. It is shown that, depending on the comparison criteria the proposed approaches can have advantages and disadvantages. In the general case, it can be concluded that these approaches are comparable by efficiency value (in terms of parameters pair - quality and computational complexity). This fact allows the developer of a particular signal or image processing system to choose the approach that is convenient and/or familiar to him. Conducted in the paper experiments show, that the proposed approaches have convincing advantages over a typical "best" way to solve the model digital image analysis/representation problem (in terms of parameters pair - quality and computational complexity).

Further research will be related to the following:
- development of alternative ways to introduce efficient linear local features;
- development of numerical methods and algorithms for a quick solution of the particular (and extended particular) problem of constructing an efficient set of jointly calculated LLFs and set of independently calculated efficient LLFs.

6. ACKNOWLEDGMENTS

This work was financially supported by the Russian Foundation of Basic Research (projects: 13-07-12103-ofi, 15-07-01164-a).

7. REFERENCES

[1] Forsyth, D.A., Ponce, J. *Computer Vision: A Modern Approach.* Prentice Hall, Upper Saddle River, New Jersey, 2003.

[2] Myasnikov, V.V. Efficient Local Linear Features for Digital Signals and Images. *Computer Optics*, 31 (4), pp. 58-76, 2007.

[3] Myasnikov, V.V. Constructing efficient linear local features in image processing and analysis problems. *Automation and Remote Control*, 72(3), pp.514-527, 2010.

[4] Myasnikov, V.V. Efficient mutually-calculated features for linear local description of signals and images. In *proceedings of the IASTED International Conference on Automation, Control, and Information Technology - Information and Communication Technology*, pp.29-34, 2010.

[5] Agarwal, R.P. *Difference Equations and Inequality: Theory, Methods, and Applications.* Marcel Dekker, New York, 2000.

[6] Lidl, R., Niederreiter, H. *Finite Fields*, Second edition, Cambridge University Press, 1997.

[7] Grigoriu, M. *Stochastic Calculus: Applications in Science and Engineering,* Birkhauser, Boston, 2002.

[8] Nussbaumer, H.J. *Fast Fourier Transform and Convolution Algorithms,* Second edition, Springer-Verlag, New York, 1982.

[9] Anderson, J.A. *Discrete Mathematics with Combinatorics*, Prentice Hall, Upper Saddle River, New Jersey, 2001.

[10] Gold, B., Rader, C.M. *Digital Processing of Signals,* McGraw-Hill Book Company, New York, 1969.

[11] Labunets, V., Gainanov, D., Berenov, D. Multiparametric wavelet transforms and packets, In *Proceedings of the 11-th International Conference on Pattern Recognition And Image Analysis*, Vol. 2. pp. 52-55, 2013.

[12] Labunets, V., Gainanov, D., Berenov, D. The best multiparametric wavelet transforms, In *Proceedings of the 11-th International Conference on Pattern Recognition And Image Analysis*, Vol. 2. pp. 56-59, 2013.

Anomaly Detection Using Spectral Mismatch Between Anomaly Pattern and its Neighborhood

Denisova A.Yu.
Samara State Aerospace University
denisova_ay@geosamara.ru

Myasnikov V.V.
Samara State Aerospace University
vmyas@geosamara.ru

ABSTRACT

In this paper we present a novel algorithm for anomaly detection in multichannel images. Proposed algorithm uses spectral mismatch criterion to describe anomalous properties of small image regions. The idea behind the criterion is that the brightness of the anomalous region can't be represented as a function of pixels comprising that region. In our paper, we consider a local pattern of anomaly and its neighborhood, and we use a linear function to approximate the anomaly at each image position. In contrast to existing global and local RXD algorithms our approach allows more adaptive and noise resistant detection of anomalies. Experimental results are presented for hyperspectral remote sensing images.

Keywords

Hyperspectral images, anomaly detection, spectral mismatch criterion.

1. INTRODUCTION

Anomaly detection is one of the common tasks of digital image processing. Generally anomalies can be described as the image regions that do not correspond to normal behavior of some valuable image characteristics. The emergence of anomalous data may be caused from different reasons including some noise or registration errors, but for anomaly detection problem it is crucial to find such portions of data that correspond to real-world features or their parts that is not typical to the environ reflected on input image.

There are different definitions of term "anomaly" which are used in various applications and depend on particular data models. An comprehensive description of variety of anomaly detection tasks can be found in [Cha09a]. In this article we consider only the problem of anomaly detection for hyperspectral images. Hyperspectral images have hundreds of image channels that correspond to narrow spectral bands, that is why every pixel is presented as vector in a multidimensional space. In hyperspectral image analysis anomaly is usually considered a small image region with spectral description sufficiently different from its neighborhood's.

One of the first anomaly detection algorithms, proposed in [Ree90a] by I.S. Reed and X.Yu, was RX-detector or RXD. Anomaly measure computed in RXD is Mahalonobis distance between current pixel vector and the average pixel vector of image. Thus anomaly is defined as a pixel which distance to the average value is the largest taking into account correlation between spectral channels of image. This algorithm demonstrates good results for images with simple single signature background, but for more complex background it is not effective. This fact and also possibility of defining the term "anomaly" in different way led up to many modifications of the RX-algorithm and other new algorithms. The examples of RXD modifications and some new algorithms are considered accordingly in [Sch07a], [Soo07a] and [Mes11a], [Ban06a], [Gu08a], [Bas07a].

In accordance with the classification of anomaly detection algorithms proposed in [Bor11a] and [Bor12a] all methods can be divided into three groups:

- subspace methods, that use global dimensionality reduction transformation for all image pixels. Usually principal component analysis or singular value decomposition is used;

- local algorithms that estimate background properties of each pixel neighborhood;

- algorithms with preliminary segmentation that aim to decompose image into regions with different background properties. Anomaly detection is performed inside these regions.

Depending on the specific task one or several of the aforementioned approaches can be used. Some algorithms may include RXD as the final processing step. This fact and along with continuous development of new algorithms, which cannot be classified into groups described above (for example, graph algorithms [Mes11a] or topological algorithms [Bas07a]), shows that such classification is very subjective.

The new algorithms proposed in this paper differ from others in definition of anomaly and image model exploited in it. They use local spatial pattern of the anomaly region and its neighborhood to incorporate assumptions about anomaly's size and form. The term "anomaly" mathematically is described by spectral mismatch criterion which is an error of anomaly candidate region approximation by its neighbourhood. In first spectral mismatch anomaly detection algorithm (SMAD) it is supposed, that image can be considered as stationary random field. The algorithm uses global spectral-spatial mismatch criterion. Because stationary random field model is used, coefficients of approximation of an anomaly-candidate region by its neighborhood are assumed to be the same for every analyzed fragment. An approximation error computed using such coefficients is the value of spectral mismatch criterion at each point and is the anomaly measure in this case.

In adaptive spectral mismatch anomaly detection algorithm (Adaptive SMAD) anomaly value is defined to be proportional to approximation error, when pixels of a potential anomaly are represented by pixels of its surroundings. Approximation coefficients are computed locally for every position of anomaly spatial pattern on the image. There is also a modification of the algorithm that employs pixel normalization.

Because both of the proposed algorithms use spectral mismatch criterion to measure anomaly of the region they can be grouped into class of spectral mismatch anomaly detection algorithms.

Proposed algorithms are compared with RXD (its global and local versions) and their superiority is shown.

2. SPECTRAL MISMATCH ANOMALY DETECTORS

Spectral mismatch algorithms compute anomaly value for each location of sliding window [Soi09a], that represent anomaly region and its neighborhood pattern.

Window is divided into two regions: interior region is interpreted as anomaly candidate and exterior region is interpreted as surroundings of potential anomaly (interior and exterior pixel sets do not intersect). Mentioned pair of pixel regions sequentially passes all possible positions on image (for example, in line-by-line scanning mode) and at each position with coordinates of central window pixel (n_1, n_2) a total "anomaly" value is computed. Total "anomaly" value for window is a result of aggregation of "anomaly" values for each pixel inside interior region. Note that aggregation can be made in different ways, for example, sum, minimum, maximum, median and so on.

Let us denote $I(n_1, n_2)$ – set of interior window pixels and $J(n_1, n_2)$ – set of exterior window pixels, where (n_1, n_2) is an image coordinate of window center, see Fig.1. The ordering of pixels within interior and exterior sets is not sufficient.

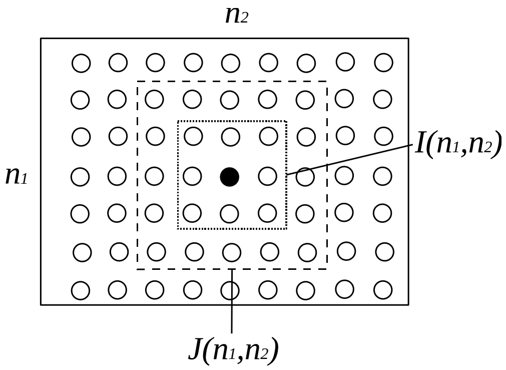


Figure 1. Interior and exterior pixel sets within processing window

Denote by v_i, $i \in I(n_1, n_2)$ and v_j, $j \in J(n_1, n_2)$ hyperspectral pixels from $I(n_1, n_2)$ and $J(n_1, n_2)$ set correspondingly.

Spectral mismatch value $\varepsilon_i^2(n_1, n_2)$ for interior pixel v_i, $i \in I(n_1, n_2)$ at window position (n_1, n_2) is defined as an error of representing interior pixel with the linear combination of pixels in exterior pixel set $J(n_1, n_2)$:

$$\varepsilon_i^2(n_1, n_2) = \left\| v_i - \sum_{j \in J(n_1, n_2)} \alpha_{ij}(n_1, n_2) v_j \right\|^2 \qquad (1)$$

where $\|\dots\|$ - some vector norm (in our case L_2 -norm), and $\alpha_{ij}(n_1, n_2)$ – coefficients of linear combination of exterior pixels that should be estimated from the image. Depending on the approach used to estimate these coefficients two algorithms can be considered.

In the first algorithm, spectral mismatch anomaly detector (SMAD), coefficients are supposed to be the same for all image. This assumption is equivalent to the following condition:

$$\alpha_{ij}(n_1,n_2)\equiv\alpha_{ij},\quad i\in I(n_1,n_2),\; j\in J(n_1,n_2)\,. \quad (2)$$

Coefficients defined in such way correspond to stationary random field image model. In this case maximum of an error Eq.1 is located at the points with sufficiently non stationary behavior.

In second algorithm, Adaptive SMAD, expression Eq.1 is used directly. It means that chosen pixel from interior set is represented as a linear combination of exterior pixels. If an error of such a representation is high, the pixel or region is interpreted as anomaly.

Below both algorithms are described and formulas for the coefficients are written.

SMAD

For spectral mismatch anomaly detector coefficients $\alpha_{ij}(n_1,n_2)$ are considered to be constant α_{ij} for all image. Their values are computed to achieve a minimum of square errors sum:

$$\varepsilon^2=\sum_{(n_1,n_2)}\varepsilon^2(n_1,n_2)\,, \qquad (3)$$

where

$$\varepsilon^2(n_1,n_2)=\sum_{i\in I(n_1,n_2)}\left(v_i-\sum_{j\in J(n_1,n_2)}\alpha_{ij}v_j\right)^2. \qquad (4)$$

Coefficients can be obtained as the solutions of the following system of linear algebraic equations:

$$\begin{aligned}&\sum_{(n_1,n_2)}v_t^T(n_1,n_2)v_k(n_1,n_2)=\\&=\sum_{j=0}^{J-1}\alpha_{kj}\sum_{(n_1,n_2)}v_t^T(n_1,n_2)v_j(n_1,n_2)\end{aligned} \qquad (5)$$

where $v_k(n_1,n_2)\in I(n_1,n_2)$ and $k=0,...,I-1$, $v_t(n_1,n_2)\in J(n_1,n_2)$ and $t=0,...,J-1$.

The coefficients α_{ij} in SMAD need to be computed only once since they are the same for each anomaly pattern position. That's why they globally define best linear approximation for all possible image regions according to required pattern. After coefficients α_{ij} have been obtained from Eq.5 for each pattern position "anomaly" value can be measured using Eq.4.

Adaptive SMAD

For Adaptive SMAD algorithm coefficients $\alpha_{ij}(n_1,n_2)$ must be different at every possible window position (n_1,n_2). These coefficients are found from orthogonal projection of chosen interior pixel vector $\overline{v}_i$ into the space linearly spanned [Kos97a] by exterior region pixels. Let us denote this projection as $\hat{\overline{v}}_i$. Then error Eq.1 will look as follows:

$$\varepsilon_i^2(n_1,n_2)=\|v_i\|^2-\|\hat{v}_i(n_1,n_2)\|^2,\quad i\in I(n_1,n_2)\,, \quad (6)$$

where $\hat{v}_i(n_1,n_2)=P_\perp^{(n_1,n_2)}v_i$ is the projection of vector-pixel v_i from set $I(n_1,n_2)$ on linear envelope of vectors from $\{v_j\}_{j\in J(n_1,n_2)}$.

Projection operator $P_\perp^{(n_1,n_2)}$ is calculated to minimize mean square error of vector v_i represented through the pixels from exterior set $J(n_1,n_2)$:

$$P_\perp^{(n_1,n_2)}=V\left(V^TV\right)^{-1}V^T \qquad (7)$$

where $V=\lfloor v_0\,v_1...v_j...v_{J-1}\rfloor$ is matrix formed from pixels from exterior set $v_j\in J(n_1,n_2)$ (to simplify formulae we will omit arguments (n_1,n_2) of a matrix below). It is evident that $\alpha=\left(V^TV\right)^{-1}V^T$. As pixels from set $J(n_1,n_2)$ can be linearly dependent among themselves, it is necessary to select a subset of linearly independent vectors or to provide projector regularization. In our work projector with regularization is used:

$$\hat{P}_\perp^{(n_1,n_2)}=V\left(V^TV+\beta I\right)^{-1}V^T\,, \qquad (8)$$

where $\beta>0$ is regularization parameter, I – identity matrix.

Total value of the spectral mismatch criterion at the current image point is evaluated as the following expression:

$$\varepsilon^2(n_1,n_2)=\sum_{i\in I(n_1,n_2)}\left(v_i-\sum_{j\in J(n_1,n_2)}\alpha_{ij}(n_1,n_2)v_j\right)^2.(9)$$

where $\alpha_{ij}(n_1,n_2)$ are the representation coefficients for current pattern position.

An optional modification of Adaptive SMAD algorithm includes preliminary normalization of all image pixels to meet the following condition:

$$\left\| v_i(n_1,n_2) \right\| = 1 . \qquad (10)$$

In this case value of error Eq.6 can be written as follows:

$$\begin{aligned} \varepsilon_i^2(n_1,n_2) &= 1 - \cos^2\left(v_i, \hat{v}_i(n_1,n_2)\right) \\ &= \sin^2\left(v_i, \hat{v}_i(n_1,n_2)\right), \quad i \in I(n_1,n_2) \end{aligned} \qquad (11)$$

where sine (or cosine) is calculated for angle between interior pixel v_i and its projection into linear subspace defined by exterior pixels. It is obvious, that error value Eq.11 unambiguously (and monotonously) depends on the specified angle.

3. EXPERIMENTAL RESEARCH

In experiments we used synthetic hyperspectral images. Images were size 256×256 pixels and 100 spectral channels corresponding to wavelengths ranging from 0.8 to 2.5 micrometer with step 0.017. Images were formed as linear combination of four "background" signatures (ACTINOLITE_AM3000, ILLITE_IL101, SEPIOLITE_SEP3101, BUDDINGTONITE_NHB2301) and two "anomaly" signatures (HEMATITE_FE2602, SIDERITE_COS2002) taken from IGCP-264 Library - CSES Beckman Spectrometer [Cla93a]. Coefficients for background and anomaly signatures were generated as stationary random fields with exponential correlation function.

Research was conducted on three synthetic images ("PIC-1", "PIC-2", "PIC-3") with correlation coefficients ρ 0.999, 0.98 and 0.45,respectively. At every image point sum of the coefficients of the linear combination was equal to one and coefficients were nonnegative. Test images were generated according to hyperspectral data linear mixture model described in [Cha02a], [Cha13a], [Cha07a]. Anomalies embedded into images were square plates with size 7×7, 5×5 and 3×3 pixels. The examples of test images with built in anomalies are shown in Fig. 2. First two images were used to compare performance with global and local RXD without dimensionality reduction. First image illustrates situation with simple constant background, the second one has more complex background.

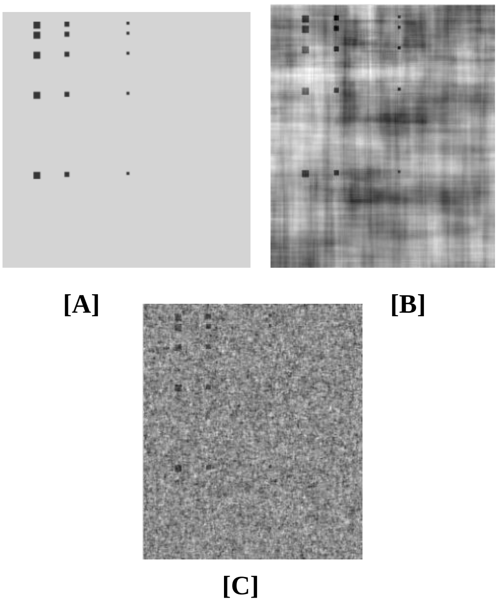
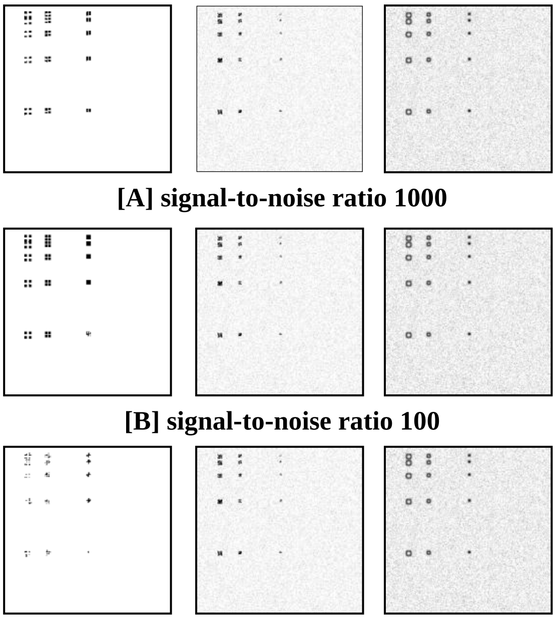


Figure 2. Examples of test images: [A] "PIC-1", [B] "PIC-2", [C] "PIC-3".



[A] signal-to-noise ratio 1000

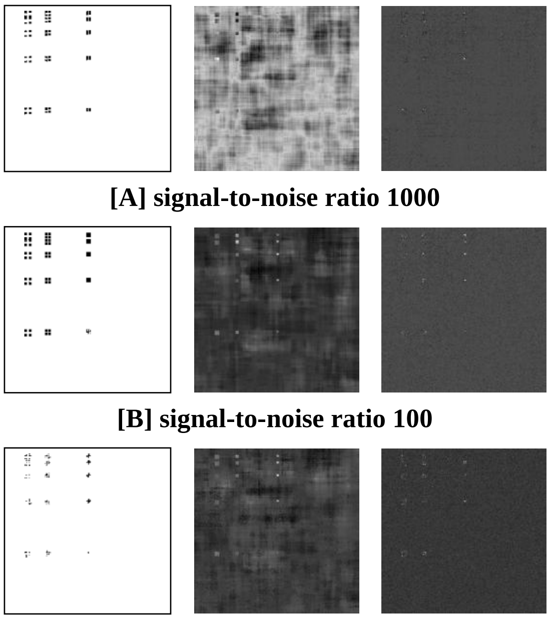
[B] signal-to-noise ratio 100

[C] signal-to-noise ratio 10

Figure 3. Experimental results for "PIC-1". From the left to right: SMAD, global RXD, local RXD (5×5 window size)

To compare algorithms the following experiment was done. At every test image channel additive independent zero-mean white noise with gauss distribution was added. Images with added noise were processed independently by two RXD modifications and proposed SMAD algorithm. Square window pattern of 5×5 pixel size was used with square interior region of 3×3 pixel size. The result of processing is shown in Fig. 3 and Fig. 4, signal to noise ratios for images were 1000, 100 and 10. Dark pixels correspond to high values of spectral mismatch value and as consequence "anomaly" region.

As we can see from Fig.3 both algorithms performs well for simple background which is close to constant (correlation coefficient is 0.999). But it is required PCA transformation before RXD to avoid fluctuations arising from RXD processing of image "PIC-1". For the experiment shown on Fig. 3 the results of RXD algorithm and its modification were



[A] signal-to-noise ratio 1000

[B] signal-to-noise ratio 100

[C] signal-to-noise ratio 10

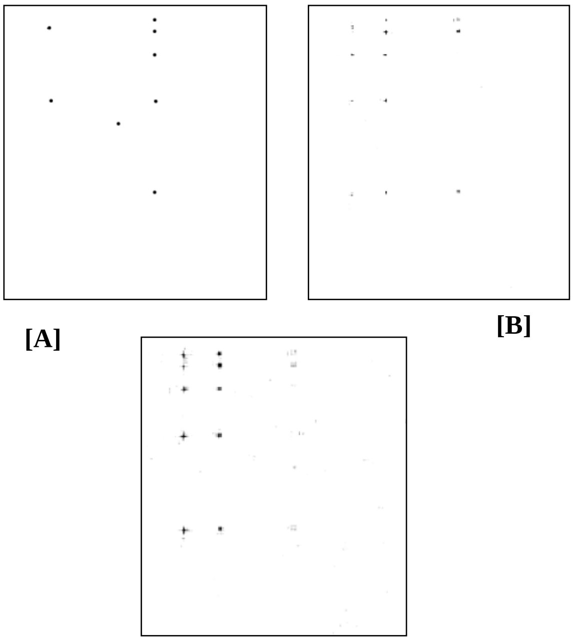
Figure 4. Experimental results for "PIC-2" From the left to right: SMAD, global RXD, local RXD (5×5 window size).

obtained for first two principal components of PIC-1 image. As for SMAD algorithm it does not require preliminary PCA transformation.

It can be seen from the results shown in Fig. 4 that for more complex background with correlation coefficient 0.98 SMAD works significantly better than RXD. For this example RXD didn't detect any anomalies while SMAD marked all of them. Thus SMAD is very noise resistant and detects anomaly from complex background better than RXD (see the results for "PIC-2").

Experiment with "PIC-3" shows the influence of the parameter selection on SMAD result. The result of processing a square window pattern with square interior anomaly-candidate region using SMAD algorithm is shown in figure 5. Sizes of window and its interior region in pixels were, respectively, 5×5 and 3×3, 7×7 and 5×5, 9×9 and 7×7. So we can see that bigger anomaly size is detected better with larger window pattern. It should be noted that window size becomes more critical parameter for images with low correlation, for "PIC-3" correlation coefficient was 0.45.

The example of Adaptive SMAD detection for image "PIC-1" with signal-to-noise ratio 100 is shown in Fig. 6. Regularization parameter was set to $0{,}01\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of matrix $V^T_{(n_1,n_2)}V_{(n_1,n_2)}$.



[C]

Figure 5. - SMAD results for "PIC-3" with window and interior region sizes respectively: [A] 5×5 and 3×3, [B] 7×7 and 5×5, [C] 9×9 and 7×7.

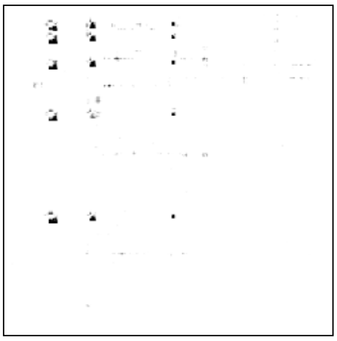


Figure 6. Adaptive SMAD result for "PIC-1". Window size and interior region size respectively 5×5 and 3×3.

Apparently, the Adaptive SMAD algorithm also yields significantly better results than RXD algorithm, although (unlike SMAD) it doesn't assume any model of the image. Adaptive SMAD has some disadvantages compared to SMAD algorithm. It is computationally expensive and generally the projection operator used in it is unstable and requires regularization. So the practical use of Adaptive SMAD algorithm has certain difficulties.

Figure 7 illustrates an example of using spectral mismatch algorithms and RXD modifications for real hyperspectral remote sensing image. We used AVIRIS Moffett field image, one of its spectral bands is shown in figure 7[A]. AVIRIS has about two

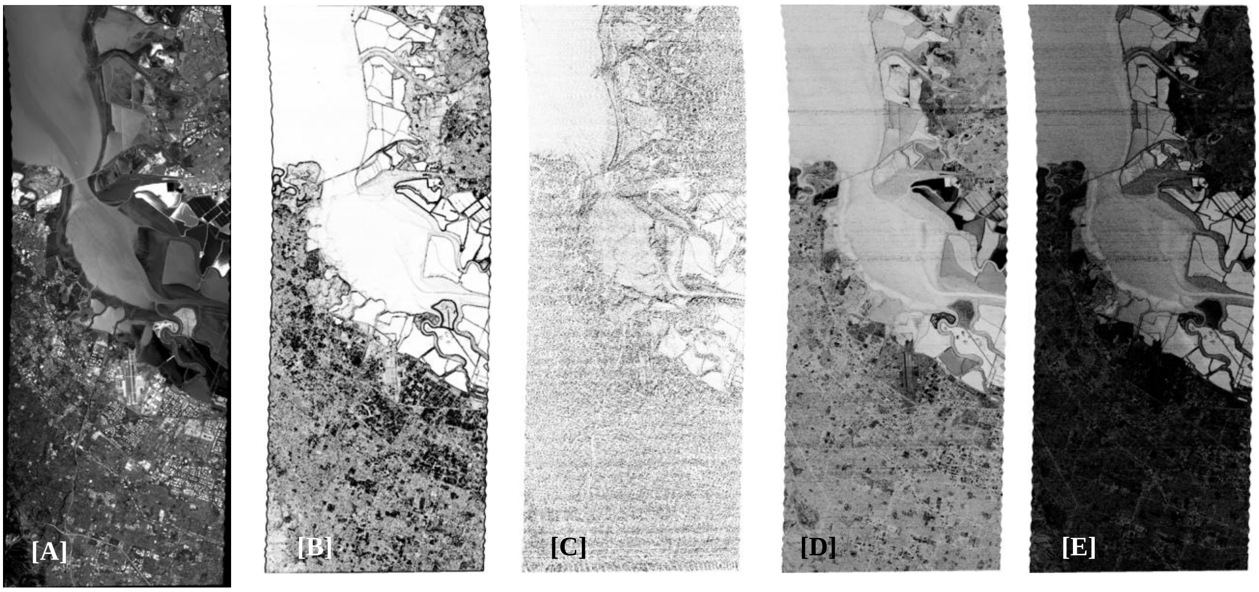


Figure 7. The results for AVIRIS Moffett field image for different anomaly detectors:[A] Original image, 550 nm band, [B] SMAD, [C] Adaptive SMAD, [D] global RXD, [E] local RXD (5×5 window)

hundred spectral channels from 400 to 2500 nanometers, some of this bands has significant noise. In our experiments we used all spectral bands of image including corrupted by noise bands. The same object in different spectral bands may look differently because of reflectance properties of its material. That is why in some spectral bands it may disappear or appears no contrast in some spectral bands. For example, in figure 7[A] bridge over the river has low contrast with water.

Fig.7[B]-[E] shows the results of all anomaly detection techniques, the darker pixels are more anomalous than the lighter. Spectral mismatch algorithms were used with 5×5 pixels square window with 3×3 interior region. Local RXD algorithm had window size 5×5. As we can see, proposed SMAD algorithm underlines borders of objects as anomalies. So the key characteristic for this algorithm is difference of spectral signatures between image objects. This fact allows algorithm to discriminate one image object from another or from background. Global RXD algorithm identified as anomalies objects which brightness was mostly different from the average brightness of the image. It is too weak condition, and we can see that only white in original image 7[A] objects were detected as anomalies by global RXD.

As for Adaptive SMAD algorithm, it demonstrates effective detection relief features in the river basin. Because of small anomaly pattern such objects as buildings were not detected as anomalies. Local RXD algorithm detected entire river bank as anomalous region that seems to be incorrect or and it makes further analysis too difficult.

It should be noted that proposed algorithms are less affected by noise than RXD. For example, on both RXD images a noise stripe can be seen in upper part of image, this stripe is absent for spectral mismatch detectors results.

4. CONCLUSION

Two new algorithms were presented in the paper, namely, spectral spatial mismatch anomaly detector (SMAD) and adaptive spectral mismatch detector (Adaptive SMAD). Their performance was studied on synthetic and real hyperspectral remote sensing images. The results of experimental comparison with basic global and local RXD algorithms were presented and advantage of proposed methods was shown.

A short comparative analysis was also provided. particularly, it has been shown that SMAD is a noise resistant algorithm and allows confident detection of anomalies on images holding on stationary random field model even in case of low signal-to-noise ratio. Adaptive SMAD algorithm has no limitation due to the absence of any underlying image model but is more computationally expensive and requires regularization parameter selection. Proposed detectors were shown to be more effective than RXD.

5. ACKNOWLEDGEMENTS

This work was financially supported by the Russian Scientific Foundation (RSF), grant no. 14-31-00014

“Establishment of a Laboratory of Advanced Technology for Earth Remote Sensing”.

6. REFERENCES

[Ban06a] Banerjee, A., Burlina, P. and Diehl, C. A support vector method for anomaly detection in hyperspectral imagery. Geoscience and Remote Sensing, IEEE Transactions on. V. 44(8), pp. 2282-2291, 2006.

[Bas07a] Basener, D., Ientilucci, E. and Messinger, D. W. Anomaly detection using topology. Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII, SPIE, V. 6565, 2007.

[Bor11a] Borghys, D., Achard, V., Rotman, S.R., Gorelik, N., Perneel, C. and Schweicher, E. Hyperspectral anomaly detection: A comparative evaluation of methods. General Assembly and Scientific Symposium, 2011 XXXth URSI, pp. 1-4, 2011.

[Bor12a] Borghys, D., Kasen, I., Achard, V. and Pernee, C. Hyperspectral Anomaly Detection: Comparative Evaluation in Scenes with Diverse Complexity. Journal of Electrical and Computer Engineering. V. 2012, pp. 16, Article ID 162106, 2012.

[Cha09a] Chandola, V., Banerjee, A. and Kumar, V. Anomaly detection: A survey. ACM Computing Surveys (CSUR), V. 41(3), pp. 72, 2009.

[Cha02a] Chang, C.I. and Shao-Shan, C. Anomaly detection and classification for hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing, V. 40(6), pp. 1314-1325, 2002.

[Cha13a] Chang, C.I. Hyperspectral Data Processing: Algorithm Design and Analysis. John Wiley & Sons, 1164 p., 2013.

[Cha07a] Chang, C.I. Hyperspectral data exploitation: theory and applications. Wiley-Interscience, 456 p., 2007.

[Cla93a] Clark, R. N., Swayze, G. A., Gallagher, A. J., King, T. V. V. and Calvin, W. M. The U. S. Geological Survey, Digital Spectral Library: Version 1: 0.2 to 3.0 microns, U.S. Geological Survey Open File Report 93-592. 1340 p, 1993.

[Gu08a] Gu, Y., Liu, Y. and Zhang, Y. A selective KPCA algorithm based on high-order statistics for anomaly detection in hyperspectral imagery. Geoscience and Remote Sensing Letters. V. 5(1), pp. 43 -47, 2008.

[Kos97a] Kostrikin, A. I., and Manin, I. I. Linear algebra and geometry. Gordon and Breach Science Publishers, p. 313, 1997.

[Mes11a] Messinger, D. W. and Albano, J. A graph theoretic approach to anomaly detection in hyperspectral imagery. Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on. pp. 1-4, 2011.

[Ree90a] Reed, I.S. and Yu, X. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. IEEE Transactions on Acoustics, Speech, and Signal Processing. V. 38(10), pp. 1760–1770, 1990.

[Sch07a] Schaum, A. P. Hyperspectral anomaly detection beyond RX. Proceedings of the SPIE Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery XII. V. 6565, 2007.

[Soi09a] Soifer, V. A. Computer Image Processing, Part II: Methods and algorithms. VDM Verlag, p.584, 2009.

[Soo07a] Soofbaf, S. R., Fahimnejad, H., Valadan Zoej, M. J. and Mojaradi, B. Anomaly detection algorithms for hyperspectral imagery Proceedings, Remote Sensing and Image Processing, Presented at the Map of the World Forum. pp. 1-8, 2007.

Visual Encoding of Automatic Identification System (AIS) Data for Radar Systems

Philipp Last
Institute for Maritime Simulation
UAS Bremen
Werderstr. 73
Germany, 28199 Bremen, Bremen
philipp.last@hs-bremen.de

Martin Hering-Bertram
Institute of Informatics and Automation
UAS Bremen
Flughafenallee 10
Germany, 28199 Bremen, Bremen
martin.hering-bertram@hs-bremen.de

Thomas Jung
Institute for Maritime Simulation
UAS Bremen
Werderstr. 73
Germany, 28199 Bremen, Bremen
thomas.jung@hs-bremen.de

Lars Linsen
Computer Science and Electrical Engineering
Jacobs University Bremen
Campus Ring 1
Germany, 28759 Bremen, Bremen
l.linsen@jacobs-university.de

ABSTRACT

The Automatic Identification System (AIS) is a maritime system mostly used for automatically exchanging tracking and other relevant information between vessels. It supports decision making of nautical personnel such as master mariners. AIS data are multivariate including many aspects for identification and localization of ships and for navigation. However, during navigation not all AIS data are made visually available to the nautical personnel. In this paper, we propose a glyph-based visualization consistent with currently used encodings for intuitively and effectively encoding further so far missing AIS data attributes on radar screens. Proposed extensions aim at increasing maritime safety by helping mariners to assess traffic situations. We applied our visualization methods to real-world data recorded at the German North Sea coast and evaluated them with the help of an expert group.

Keywords

Information visualization, Visualization techniques and methodologies, Application

1. INTRODUCTION

The Automatic Identification System (AIS) allows for transmitting data between AIS systems, which can be installed on vessels, base stations like harbor authorities, landmarks like buoys, or on search and rescue airplanes. The AIS data which are exchanged is divided in three different types [ITU13]:

- Static data (e.g., vessel name and the dimensions of the vessel)
- Dynamic data (e.g., vessel position, course over ground, and heading)
- Voyage-related data (e.g., current draught, description of cargo, and destination)

Thus AIS is a useful complement to systems like Radio Detection and Ranging (radar) by providing additional information which would otherwise not be available. Both static and dynamic AIS data provide useful information for course corrections and collision avoidance, respectively.

Radar systems which are installed on vessels make use of received AIS data by adding additional information extracted from the AIS data stream to the radar screens. So far the most common way of displaying AIS information is a visual encoding of basic information such as the geographical position and the current heading of the vessel. However, AIS data provide much more information and therefore the potential of AIS data for navigational purposes is not yet fully exploited.

We extend the existing AIS glyphs through identifying and encoding additional relevant AIS data attributes while considering general glyph design principles. Our main contributions are:

- Summarizing the current state of the art of representing AIS data on radar screens.
- Developing a visual encoding of additional identified attributes with the help of maritime experts which builds on and extends existing glyphs to ensure a high acceptance by users.
- Evaluating our proposed results by collecting feedback from an expert group.

AIS data are also used within Electronic Chart Display and Information Systems (ECDIS). Even though there is a strong link between both radar and ECDIS our focus lies on displaying AIS data on radar screens, i.e., on devices with limited resolution and with low rendering performance.

2. RELATED WORK

Currently, within the visualization area AIS data are used to predict and visualize vessel movements. Within this context important work has been released by Scheepens et al. who created interactive density maps or contour based visualizations of vessels and vessel trajectory data by using AIS data [Sch11a-c] [Sch14]. However, the AIS data representation as glyphs used by mariners on board has not advanced in the past years. A glyph is a small visual object which represents attributes of a data record. A variety of design guidelines and design criteria exist to develop glyphs [Che12] [Mag12][Pet10][Rop11].

Within this context important work related to AIS data has been released by Motz et al. [Mot08] who performed an experimental investigation for the German Federal Ministry of Transport, Building, and Housing to evaluate the presentation of AIS target information on Electronic Chart Display and Information Systems. They state that "[…] there is a compelling need for a suitable graphical presentation of AIS information in order to improve target identification, to reduce the mariner's workload by presenting information in a readily assimilated format, to enhance 'Situation Awareness', and thereby to reduce the risk of collision and to improve the safety of navigation, particularly in congested waters." [Mot08]. Further work has been performed by Motz and Widdel by evaluating the graphical presentation of AIS information on ships [Mot01]. Two experiments were conducted with simulated traffic scenarios on ECDIS and radar systems to identify symbols including symbol properties and visual channels such as size and color which are most suitable to display AIS information. Their results show that oriented triangles with additional attributes are the most suitable glyphs to represent vessels even though a diamond shaped symbol caused a faster detection rate of moving vessels [Mot01]. Based to the work of Motz and Widdel, guidelines for the presentation of navigation-related symbols have been released by the International Maritime Organization in 2004 [IMO04] which are shown in Fig. 1.

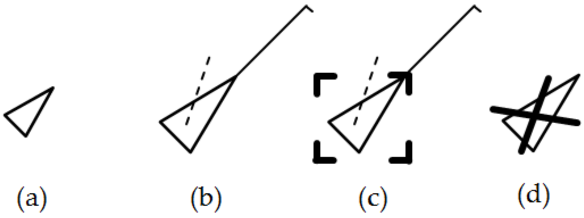


Figure 1. AIS symbols representing different AIS targets as recommended in [IMO04].

Fig. 1 (a) shows a non-moving AIS target symbol indicating the current position and heading of a vessel, (b) shows the recommended symbol for an active AIS target showing a rate of turn indication (ROT) as a small flag connected to a line which emphasizes the heading (HDG). The dashed line is a vector consisting of speed over ground (SOG) and course over ground (COG) and represents the actual movement and time based course prediction of the vessel which may differ from the HDG information. Glyph (c) represents a selected target and (d) a lost target which means that no AIS message has been received from this entity for a specific amount of time. Dangerous targets should be drawn bold and colored red. In addition they should be flashing until they are acknowledged. In 2008 an amendment to these guidelines had been released [IMO08]. AIS Search and Rescue Transmitters (AIS-SART) can be identified by their unique Maritime Mobile Service Identity (MMSI) number, therefore the latest update [IMO08] contains an additional glyph indicating AIS-SART targets, see Fig. 2.

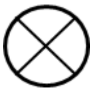


Figure 2. Recommended symbol for AIS-SART as shown in [IMO08].

Further general design considerations with respect to maritime data can be adopted from the guidelines released by the International Hydrographical Organization (IHO) [IHO10]. Since the IHO intends to ensure a clear and unambiguous display of ECDIS screens, the proposed specifications [IHO10] are considered within our glyph design. In conclusion the current glyphs used for the representation of AIS targets shown in Fig. 1 and Fig. 2 give an indication to the mariner whether an AIS information is available or not. This includes:

- Geographical position consisting of latitude and longitude,
- HDG,
- COG,
- SOG, and
- ROT not equal to zero.

Therefore mostly visual channels of geometric and topological/relational type are used to encode AIS data visually. E.g., the guidelines for the presentation of navigation-related symbols almost do not make use of further visual channels such as color or transparency even though current radar systems provide color support. Furthermore, not every AIS system transmits all of the mentioned data fields as shown in [Las14]. E.g., it is possible that a vessel does not transmit HDG and COG, i.e., it is difficult to draw the triangle symbol correctly rotated. In contrast even more information might be available for a specific vessel but is not yet visually encoded in the glyphs in Fig. 1 and 2. This includes the ship type or the draught of a vessel. This leads to a lack of AIS indicators and missing glyphs in specific situations.

3. LIMITATIONS OF CURRENT AIS REPRESENTATION

The current graphical representation of the information provided by AIS covers a wide range of aspects relevant for navigational purposes. However, while evaluating recorded AIS data, we identified that the current visual encoding of AIS data is in some traffic situations not sufficient to display all relevant information. We address this problem by giving examples for such situations as well as by making proposals to extend existing symbols as well as by adding new symbols for currently not covered aspects.

Vessel type encoding

As shown in Fig. 2 AIS-SART systems can be easily identified on a radar screen since they are displayed with a separate symbol. However, AIS systems are installed on many more vessel types. Examples taken from [ITU13] are *Pleasure Craft, High Speed Craft, pilot vessels, law enforcement vessels,* or *Cargo*. This information is not encoded within the current AIS symbol set. However, encoding the vessel type allows a mariner identifying vessels which are relevant for the current situations at sea. The vessel type gives information about a vessel's maneuverability and may additionally include a cargo classification. Encoding additional vessel types allows the mariner to distinguish faster between radar and AIS echoes in situations with heavy traffic. Therefore it allows them to get in contact with, e.g., a Search And Rescue (SAR) vessel. Indicating the vessel type also allows a manual prediction of possible vessel movements, since a high speed craft has a bigger operational radius than a tanker and can also quickly change its movement direction. We propose to extend glyph (b) in Fig. 1 by adding a transparent filling if the vessel has transmitted its vessel type. In addition, we propose to use different colors to encode specific groups of vessels.

Navigational status encoding

In total, 16 different navigational statuses exist, of which seven are reserved for future use [ITU13]. The navigational status also belongs to the static information. The statuses *At anchor, Moored,* and *Aground* are currently concluded as non-moving AIS targets with the appropriate glyph shown in Fig. 1. All remaining statuses are considered as moving AIS targets. Related to the navigational status, our dataset shows that within crowded situations, e.g., in harbors, non-moving AIS targets may clutter the screen. However, filtering non-moving targets is not always possible since one may be interested in data of such a vessel.

Dimensions encoding

The AIS system provides the possibility to transmit the dimensions of a vessel. The dimensions are static, since they are entered manually when the AIS system is initially configured. The approach of displaying these dimensional values is described as AIS Target – True Scale Outline in [IMO04]. It is written that "A true scale outline may be added to the triangle symbol. […] Located relative to reported position and according to reported position offsets, beam, and length. Oriented along target's heading." [IMO04]. Even though these guidelines are almost 10 years old only few radar systems provide the possibility to show the vessel's dimension as an additional overlay. However, the vessel dimensions provide important information for collision avoiding and navigational purposes. The AIS target glyph of Fig. 1 (b) does not provide any information about actual vessel dimensions. Depending on the radar scale, the actual vessel size and also shape may be smaller or even bigger than the AIS target glyph. The radar echo itself may provide further information, however an echo is not always available and depending on the weather or other passing objects not reliable since shadowing may occur.

(SAR) aircraft encoding

Even though AIS is intended for usage by SAR aircrafts, our data set evaluation shows that it is not uncommon to install AIS systems on further aircrafts such as planes or helicopters which are, e.g., used to transfer workers to oil rigs. So far current systems do not display these aircrafts or they use the same symbol which is used for displaying vessels. This may lead to confusion since aircrafts have a different behavior since they are much faster than vessels and may not always provide a radar echo. So far no glyph representing aircrafts exists leading to irritations when aircrafts are being displayed with the vessel symbol shown in Fig. 1. Since SAR transmitter are represented by an own AIS glyph, we propose using a separate glyph for aircrafts as well.

Draught encoding

So far the vessel draught which is measured in meters is not visually encoded. Encoding the draught allows the mariner to estimate possible vessel movements since a container vessel with full cargo has a bigger draught than an empty one. This information cannot be obtained from the radar echo. In addition to this the draught value gives – if available – information about possible vessel movements and restrictions. E.g., if a vessel has a large draught, it may only drive in specific fairways. Furthermore encoding the draught roughly indicates a vessel's size to the user since a container ship has usually a higher draught than, e.g., a sailing yacht.

4. CONSTRAINTS FOR DISPLAY

Current radar systems which are used on board of professional operating vessels consist of a radar antenna to generate radar echoes, a radar processor unit (RPU), a radar screen to display radar echoes and further information calculated by the RPU and a trackball as an input device allowing the user to interact with the system. The RPU is usually an embedded system which queries and processes current sensor states and has therefore a restricted performance. Within the professional operating field a common size for radar screens is 19" with a resolution of 1280x1024 (SGXA). Beside the trackball further buttons exist which are connected to specific functionalities which must be accessed quickly. Those buttons are also related to the AIS data visualization, e.g., switching the AIS visualization on or off. Fig. 3 shows a radar screen excerpt displaying a situation with and without AIS overlay activated.

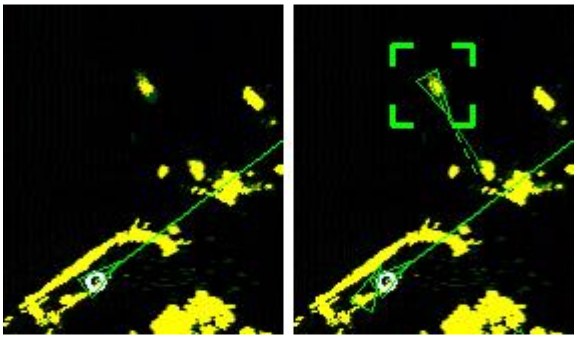


Figure 3. Radar screen excerpt. The white point indicates the own vessel position.

Visual encoding techniques are often used to encode rich data sets which consist of recorded data. Appropriate hardware and tools are available for data processing. However, Fig.3 shows that a radar system is a real time application with a restricted user interface and it is obvious that a graphical representation has to be very basic. Glyphs are represented by a limited number of pixels due to the low resolution and smoothly shaded objects cannot be rendered due to low-level graphics hardware. Extending the existing AIS encoding must not distract the user from his/her tasks. The visual encoding of AIS data must cause minimal occlusion of radar information while providing an additional benefit for the mariner. Large and complex glyphs are not an option and official guidelines such as [IHO10] must necessarily be considered. These guidelines "[…] ensure a base […] and appropriate compatibility with paper chart symbols as standardized in the Chart Specifications of the IHO" [IHO10]. Furthermore it is necessary to ensure that "[…] the display is clear and unambiguous" and "[…] that there is no uncertainty over the meaning of colors and symbols on the display […]". [IHO10] also includes technical limitations. Furthermore extensive studies have been performed to identify the most suitable glyphs shown in Fig. 1 and Fig. 2 to represent AIS data attributes. For that reason the concept of familiarity as described in [McD99] should be considered and therefore AIS extensions should be based on the existing encoding which has been proven and tested for years to achieve high acceptance by users. In addition to this, while developing AIS glyphs one has to consider that AIS data is sometimes partially missing or simply wrong [Har07][Las14].

5. GLYPH-BASED VISUAL ENCODING

Within this section we are presenting our novel glyph-based visual encoding approach. The existing visual AIS encodings are extended to overcome the problems mentioned in Section 3 while considering the constraints of Section 4. The evaluation is done by performing a qualitative user study with domain experts. The group of domain experts consisted of five experts aged between 35 and 65. All experts studied at a maritime academy, are (master) mariners, and have been working continuously in the maritime area. The range of professional experiences varies between 10 and 45 years. These experts are referred below as Expert 1, Expert 2, … , Expert 5.

Vessel type encoding

While experimenting with different colors and questioning users of our ship handling simulator we evaluated that, even though in total 100 codes exist to describe different vessel types, only a few of them are of interest for navigational purposes. We introduced two additional colors to indicate if a vessel belongs to a specific group. Perceptual studies have shown that the number of colors to be used shall be restricted [Hea96], here we suggest not using more than four different colors in total due to the limited resolution.

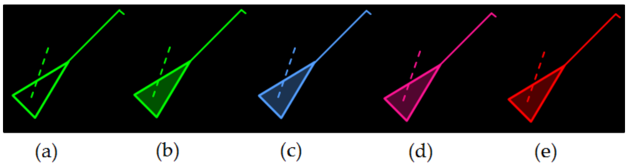


Figure 4. Encoding the vessel type information by using transparent fillings and two additional colours (blue and magenta).

Fig. 4 shows our proposed results using color and transparency as visual channels. Symbol (a) is equal to the AIS symbol for active targets shown in Fig. 1 (b) which is represented in a bright green RGB(0,255,0) by almost all current radar systems. We propose to use this glyph if no information about the vessel type is available (yet). Symbol (b) indicates that the ship type information is available and has been received. However, the ship type is not relevant for navigational purposes and therefore not separately color-coded. Examples for symbol (b) are the following ship types taken from the official AIS standard: *Local Vessel*, *Reserved*, *Pleasure Craft*, *Sailing*, or *Other Type*. If desired by the user the detailed ship type can be obtained from the radar menu. Symbol (c) uses a desaturated blue such as RGB(84,159,255) to represent assistance vessels like pilots and tugs, since blue as a foreground color is currently not used for AIS representations [IHO10]. Symbol (d) indicates that the AIS target represents an official vessel such as SAR vessels and Law Enforcement vessels using desaturated magenta color such as RGB(255,20,147), since magenta "[…] is used to highlight critically important features[…]" [IHO10]. Desaturated colors are used since the usage of saturated colors resulted in undesired pop-up effects. This pop-up effect should be reserved for dangerous targets which are being displayed red. Used filling colors are equal to the border colors, however the main body of the triangles is filled using transparency whereas the triangle borders are solid lines. Evaluations showed that a transparency value of around 68% allows the user to identify the color as well as to display radar echoes which are lying underneath the drawn AIS glyph. It is possible to use only colors for indicating the vessel type since each mariner has to pass a fitness test for sea service within regular intervals, starting with the beginning of the nautical education. Therefore mariners are tested for color blindness and similar diseases which represent a criterion for exclusion.

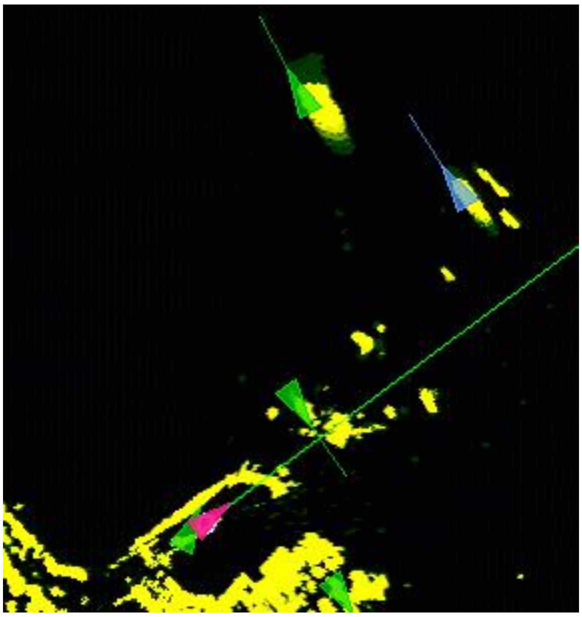


Figure. 5. Extended AIS glyphs using additional colors indicating the vessel type while using real data.

Discussion. While developing the glyphs we tried to group further vessel types such as *Tanker*, *Cargo*, and *Passenger* to a common group cargo or to display the vessel type *High Speed Craft* with an additional color. However, while experimenting with grouping and displaying further vessel types we realized that only few suitable colors with high contrast to the background, to radar echoes, and to AIS targets exist. Furthermore users of our ship handling simulator reported that coloring further vessel types beside (c) and (d) does not provide an actual benefit while navigating. We identified the same for the encoding of *Hazardous categories A-D* which can be added to the vessel type. E.g., it is possible to declare a vessel type as *Cargo - Hazardous category C* while using AIS. One approach was to encode an eventually available hazardous category by using the color red RGB(255,0,0) for the solid triangle border while still using the proposed main body colors in Fig. 4. This caused a pop-up effect as described by Chung et al. [Chu13]. Even though this implementation provides a good visual interpretability, users of the ship handling simulator reported that the benefit of encoding hazardous categories is not significant for navigating.

Results. Fig. 5 shows our proposed encoding using recorded AIS and radar data. We can observe that all visible vessels transmitted their vessel type. The appropriate radar echoes are still visible since transparency is used. Furthermore it is visible that the blue target is an assistance vessel whereas the magenta target is a SAR vessel, more precisely, the SAR vessel which had been used to record the shown radar and AIS data. Since we suggest assigning a higher priority to SAR and Law Enforcement vessels as shown in Fig. 4 (d) these vessel types should always be drawn on top. This approach allows identifying and selecting such AIS targets even in cluttered situations.

Evaluation. The evaluation feedback from the expert group is positive. Expert 3 states that the usage of colors to represent vessel types "[…] is definitely a huge advantage". Expert 3 also agrees that the amount of color groups which were developed within our work represent the maximum. Expert 2 agrees that our color encoding is helpful. Furthermore Expert 2 states that a further differentiation of vessel types with additional colors would be confusing. Only Expert 4 stated that he would not color any of the different vessel types at all, since, despite the usage of transparency, radar echoes might be covered if too many vessels are located close to each other. Concerning the vessel types only pilots are of interest for Expert 4. In summary, the expert group agrees that the vessel type encoding provides a benefit. The opinions only differ related to the vessel types which should actually be encoded. Examples are Expert 5 who is interested in a separate encoding for high speed crafts and Expert 4 who is only interested in pilot vessels.

Navigational status encoding

We used real-world data to analyze different situations with non-moving and moving targets. Since the current glyph for non-moving targets is still similar to the glyph for moving or active targets, evaluations showed that it can be difficult to distinguish the two glyphs, especially in areas with a high vessel density. Thus, we suggest a more meaningful glyph for non-moving targets as shown in Fig. 6.

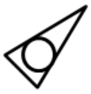


Figure 6. Indicating non-moving vessels by drawing a circle inside of the triangle.

Results. Fig. 7 shows a scenario with the proposed glyph visually encoding non-moving vessels. Furthermore the proposed vessel type encoding is visible. The radar range is 1.5 nautical miles and the images are using a reduced scale. All non-moving targets can be distinguished from active targets even though the images show a cluttered scene representing real data.

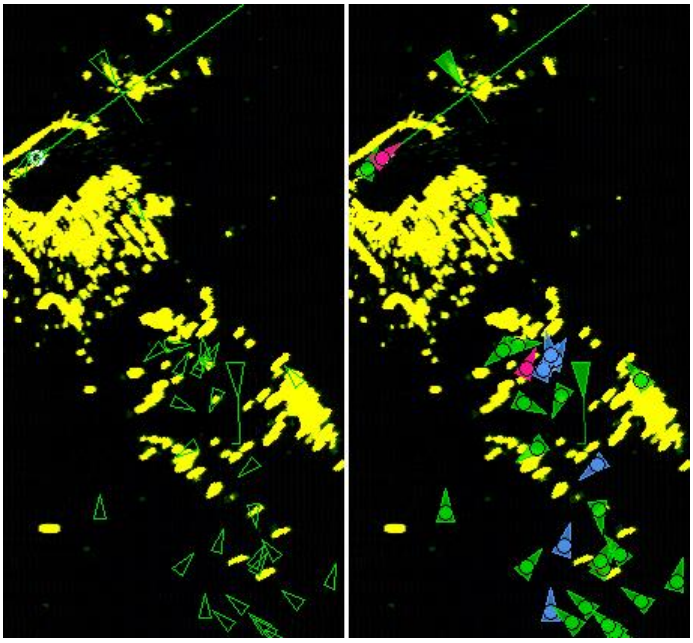


Figure. 7. Comparison of currently used AIS glyphs (left) and proposed glyphs (right) concerning vessel types and navigational status.

Evaluation. Only Expert 4 stated that the currently used encoding is sufficient to display non-moving targets. All other experts agreed that our proposed encoding allows for a faster assessment of the scene and to distinguish non-moving and active targets. E.g., Expert 2 stated that our encoding is "[…] reasonable and allows for a faster situation assessment".

Dimensions encoding

Since only 3.4% of all vessels fail to transmit their dimensions [Las14], current ECDIS and radar system should support their visual encoding. The guidelines for the presentation of navigation-related symbols recommend drawing a true scale outline [IMO04]. However, during our evaluations we observed that (depending on further visual channels such as color) a vessel's outline is hard to spot even on low radar ranges. Our evaluations showed that drawing the dimensional values is best recognizable when being drawn as a filled polygon with a slightly differing border color. We recommend a cyan filling color of RGB(0,255,255) and a blue outline color of RGB(0,0,255). The polygon itself should be drawn on top of the appropriate AIS target symbol, since it provides more detailed information and is easier to spot as shown in Fig. 8. One target to the left has not transmitted its dimension values indicating that these values might not be available or simply have not been received so far.

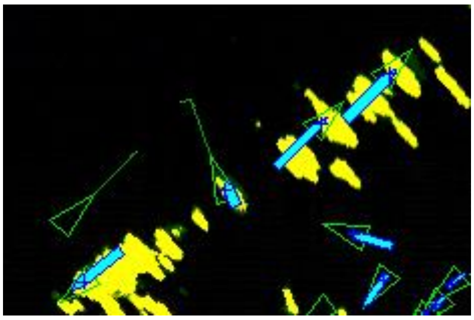


Figure 8. Dimensions are drawn as additional overlay for AIS targets. Radar range is 1.5 nautical miles.

Results and Discussion. As exemplarily shown in Fig. 8, the AIS glyph size is barely ever similar to the actual dimensions of the visible vessels. Only using the triangle glyph may cause a wrong impression to the mariner. As shown in Fig. 8, several driving and moored vessels are actually located mostly outside of the AIS target glyph since only the antenna position which is in these cases close to the bow is used to draw the AIS target glyph. The antenna positions are displayed by a blue cross for test purposes. We also evaluated that the vessel's dimensions should be displayed independently of the radar range. Even though vessels with small dimensions are more difficult to spot if the radar range exceeds 1.5 nautical miles, larger vessels are still good to spot. Therefore displaying vessel dimensions should not be related to a specific radar range but able to be (de-)activated by using an additional button to avoid cluttered scenes. While working with the provided vessel dimensions, each mariner should be aware that the dimensional data are error-prone, since they were entered manually. Therefore, uncertainty of these data has to be considered, especially when vessels are close to each other. Nevertheless these data should be used since it provides useful information for mariners.

Evaluation. The expert group agrees that our proposed overlay provides a huge benefit. Nevertheless Expert 2 states that "[…] it is important not to clutter the radar screen.". Expert 3 states that "[…] the amount of features being displayed should not distract from the actual situation". Therefore all experts agreed to our decision that it should be possible for the mariner to (de-)activate this kind of overlay as and when required.

(SAR) aircraft encoding

As mentioned beforehand AIS systems can also be installed on aircrafts. While evaluating recorded data we identified fast moving AIS aircraft targets. Currently SAR Aircrafts and vessels share the same glyph for active AIS targets as they are both AIS targets. While analyzing AIS data we identified scenarios with SAR aircrafts and vessels in which using the same glyph may result in confusion, since the user expects that the active AIS glyph represents a vessel and not a SAR aircraft. We propose to use a glyph which has been initially developed for vessels by [Mot01] but has been replaced by the glyph shown in Fig. 1 (b) to avoid confusion in situations where SAR aircrafts are additionally displayed. Even though a COG attribute is included in the appropriate AIS message we suggest using the glyph shown in Fig. 9 to represent SAR aircrafts since our evaluations show that the COG attribute might also not be available.

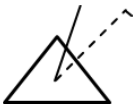


Figure 9. Proposed (SAR) aircraft glyph which can be distinguished from the active target glyph shown in Fig. 1(b) since no rotation is used. COG might be indicated by a solid line, if available.

Evaluation. The feedback we got was mixed. Expert 1 agrees that a separate symbol should be used. However, Expert 1 states that the symbol of Fig. 9 "is too similar to the current active target symbol". Expert 2 agrees with Expert 1 that a general representation is desired but the symbol of Fig. 9 might not be suitable. All further experts state that civil aircrafts should be in general not displayed since they are not of interest. However, since it is not possible to distinguish between civil and SAR aircrafts because of the used AIS message type, a usage of a separate glyph is meaningful.

Draught encoding

Visually encoding the dimensions allows for predicting a vessel's route and possible maneuvers which can be performed by the vessel. The same applies for a vessel's draught, which has not been visually encoded so far. We propose to distinguish the three classes *small draught* of 0m to 2m, *middle draught* of 2m to 10m, and *large draught* of more than 10m and visually encode this information with 1, 2, or 3 filled circles at the beginning of the heading line, see Fig. 10. Missing circles indicate that there is no draught information available. If a vessel is a non-moving target, both heading line and draught information are not displayed since only moving targets whose courses are related to the own vessel's course are of interest to a mariner in terms of navigation. Therefore draught information for non-moving targets does not provide any benefit.

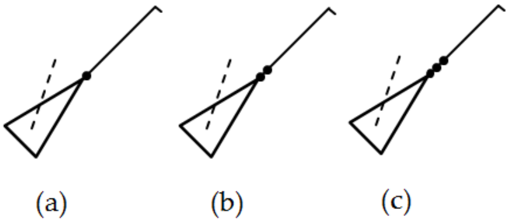


Figure 10. Encoding draught values using the classes of draughts represented by filled circles attached to the heading line.

Results and Discussion. Fig. 11 shows exemplarily the proposed encoding applied to real world data. The left image of Fig. 11 shows two vessels with a middle draught and a few non-moving targets. The right image of Fig. 11 shows a vessel with a large draught. In the present case, the radar echo already indicates that this vessel has a huge size and therefore a higher draught. However, the radar echo may not always be available. E.g., the vessel with a middle draught in the right image is almost completely shadowed by the bigger vessel and has therefore almost no radar echo. Encoding both dimensional and draught values as shown above allows mariners to assess traffic situations and to predict possible vessel movements.

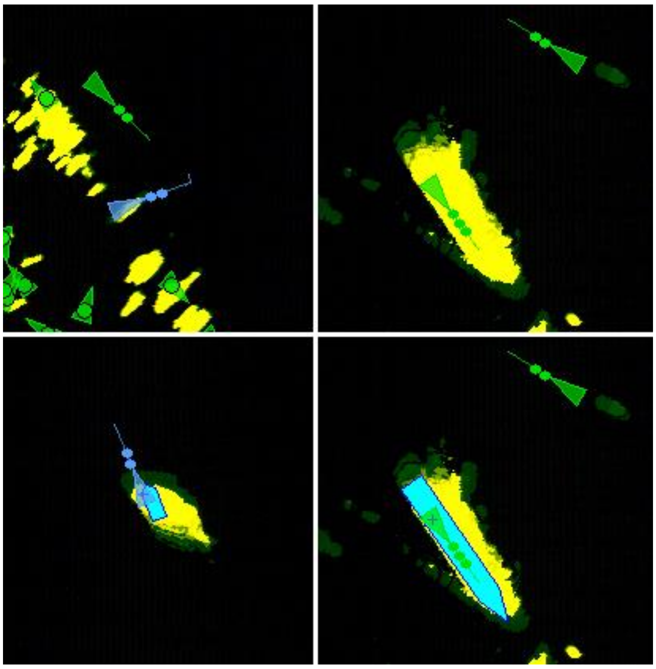


Figure 11. Encoding AIS draught values.
Top left: Two vessels with middle draught (2m-10m) can be spotted. Top right: One vessel has middle draught; one vessel has large draught with more than 10m. The bottom images show the dimensions additionally (if received).

Evaluation. Expert 3 and Expert 4 would slightly modify our proposed values. These experts would only distinguish between heavy draught bigger than 10m and no draught. However, Expert 5 prefers the proposed values. In general, all experts agreed that our proposed draught encoding is a huge benefit for the mariner. E.g., Expert 2 stated that “[…] a faster assessment of possible vessel movements and movement restrictions is possible” when using our encoding. Expert 5 states that our encoding is “[…] absolutely meaningful, especially in narrow waters”.

6. CONCLUSION & FUTURE WORK

Within this paper we identified several AIS aspects which provide a benefit for users of radar systems and which are currently not visually encoded. We proposed several extensions for using glyphs to encode this information visually on radar screens. While identifying and encoding missing AIS attributes such as draught, each extension represents a trade-off between encoding data as detailed as possible and not overloading the radar screen. While implementing and evaluating different approaches we considered the concept of familiarity as an important factor. Therefore, our work is based on current AIS glyphs. Experiments were conducted with recorded traffic scenarios on radar systems to collect expert feedback. In conclusion, all experts agree that AIS features need to be activated as and when required. If detailed information is desired an additional inspection needs to be performed by the user to avoid cluttered scenes. Furthermore our work shows that different experts assign different features a higher or lower priority. Therefore future work should include a detailed user study as well as controlled experiments to evaluate, e.g., reaction times.

7. ACKNOWLEDGMENTS

The authors are grateful for the assistance by Deutsche Gesellschaft zur Rettung Schiffbrüchiger (DGzRS) (German Maritime Search and Rescue Service). This study has been funded by the Federal Ministry of Education and Research (BMBF) (Grant no. 03FH001PX2).

8. REFERENCES

[Che12] M. Chen and L. Floridi, "An Analysis Of Information Visualisation," Synthese, vol. 190, pp. 5,6, 2012.

[Chu13] D. H. Chung, P. A. Legg, M. L. Parry, R. Bown, I. W. Griffiths, R. S. Laramee and M. Chen, "Glyph Sorting: Interactive Visualization For Multi-Dimensional Data," Information Visualization, pp. 4, 2013.

[Har07] A. Harati-Mokhtari, A. Wall, P. Brooks and J. Wang, "Automatic Identification System (AIS): Data Reliability and Human Error Implications," Journal of Navigation, vol. 60, pp. 373-389, 2007.

[Hea96] G. Healey, C., 1996. Choosing Effective Colours for Data Visualization, IEEE, ed. In: Visualization '96. Proceedings., Oct. 27-Nov. 1 1996, pp. 263-270.

[IHO10] IHO. S-52 - specifications for chart content and display aspects of ECDIS. 2014(05/05), pp. 13,14,17,38,39,40,45. 2010. Available: http://www.iho.int/iho_pubs/standard/S-52/S-52_e6.0_EN.pdf

[IMO04] IMO. Guidelines for the presentation of navigation-related symbols, terms and abbreviations. 2014(05/05), pp. 4,9. 2004. Available: http://www.iho.int/mtg_docs/International_Organizations/IMO/ECDIS-ENCDocuments/English/SN_Circ243.pdf

[IMO08] IMO. Amendment to guidelines for the presentation of navigtion-related symbols, terms and abbreviations. 2014(05/05), pp. 1,2. 2008. Available: http://www.iho.int/mtg_docs/International_Organizations/IMO/ECDIS-ENCDocuments/English/SN_Circ243-Add.1.pdf

[ITU13] ITU. Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band. 2013(05/12), pp. 3-109. 2010.

[Las14] P. Last, C. Bahlke, M. Hering-Bertram and L. Linsen, "Comprehensive Analysis of Automatic Identification System (AIS) Data in Regard to Vessel Movement Prediction," Journal of Navigation, vol. 67, pp. 1-19, 2014.

[Mag12] E. Maguire, P. Rocca-Serra, S. Sansone, J. Davies and M. Chen, "Taxonomy-Based Glyph Design—With A Case Study On Visualizing Workflows Of Biological Experiments," IEEE Trans. Visual. Comput. Graphics, vol. 18, pp. 6, 2012.

[McD99] S. J. P. McDougall, M. B. Curry and O. de Bruin, "Measuring symbol and icon characteristics: Norms for concreteness, complexity, meaningfulness, familiarity, and semantic distance for 239 symbols," Behavior Research Methods, Instruments. & Computers, vol. 31, pp. 4,5, 1999.

[Mot01] F. Motz and H. Widdel, "Graphical Presentation of AIS Information on Ships," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 45, pp. 1-6, 2001.

[Mot08] F. Motz, H. Widdel, S. MacKinnon, A. Patterson and L. Alexander, "Experimental investigation for presentation of AIS symbols on ECDIS in a motion-based ship bridge simulator," in Ergonomie Und Mensch-Maschine-Systeme, Springer Berlin Heidelberg, Ed. 2008, pp. 147-159.

[Pet10] R. Pettersson, "Information Design—Principles and Guidelines." Journal of Visual Literacy, vol. 29, pp. 6,9, 2010.

[Rop11] T. Ropinski, S. Oeltze and B. Preim, "Survey Of Glyph-Based Visualization Techniques For Spatial Multivariate Medical Data," Comput. Graph., vol. 35, pp. 2, 2011.

[Sch11a] Scheepens, R., Willems, N., van de Wetering, H., Andrienko, G., Andrienko, N. and van Wijk, J.J., Composite Density Maps for Multivariate Trajectories. IEEE Transactions on Visualization and Computer Graphics, 17(12), pp. 2518-2527, 2011.

[Sch11b] Scheepens, R., Willems, N., van de Wetering, H. and van Wijk, J.J., Interactive Density Maps for Moving Objects. Computer Graphics and Applications, IEEE, 32(1), pp. 56-66, 2011.

[Sch11c] Scheepens, R., Willems, N., van de Wetering, H. and van Wijk, J.J., Interactive Visualization of Multivariate Trajectory Data with Density Maps. Pacific Visualization Symposium (PacificVis), 2011 IEEE, pp. 147-154, 2011.

[Sch14] Scheepens, R., van de Wetering, H. and van Wijk, J.J., Contour based visualization of vessel movement predictions. International Journal of Geographical Information Science, 28(5), pp. 891-909, 2014.

3D Reconstruction of Outdoor Scenes Using Structure from Motion and Depth Data

Keisuke Fujimoto
Hitachi Ltd.
1-280, Higashi-Koigakubo,
Kokubunji-shi
185-8601, Tokyo
keisuke.fujimoto.qb@hitachi.com

Takashi Watanabe
Hitachi Ltd.
1-280, Higashi-Koigakubo,
Kokubunji-shi
185-8601, Tokyo
takashi.watanabe.dh@hitachi.com

ABSTRACT

Recently, low-cost and small RGB-D sensors appear massively at the entertainment market. These sensors can acquire colored 3D models using color images and depth data. However, a limitation of the RGB-D sensor is that sunlight interferes with the pattern projecting LED. The sensor is most suitable only for indoor scenes. Some RGB-D sensors are available in outdoor scenes. However, the measurement range is limited because the light of LED spreads in all directions. In this research, we developed a novel measurement method for RGB-D sensors, which can measure shapes in outdoor scenes. This method uses several measurement data from multiple viewpoints, and estimates the shape and the sensor poses using Structure from Motion (SfM). However, a conventional image-based SfM cannot determine a correct scale. To determine the correct scale, our method uses the depth information that is obtained from partially acquired area which is near to the viewpoints. Then, our method optimizes the shape and the poses by a modified bundle adjustment with the depth information. It minimizes the reprojection error of the features in the acquired images and the depth error between the estimated model and the measurement depth. At last, our method generates dense point cloud using a multi-view stereo algorithm. Using both the acquired images and depth data, our method reconstructs the shape which locates out of measurement range in outdoor environment. In our experiment, we show that our method can measure the range up to 20 meters away by measuring from several viewpoints in the range of 5 meters using a RGB-D sensor in outdoor scenes.

Keywords

RGB-D sensor, Structure from Motion, Bundle Adjustment, Point Cloud, Multi View Stereo

1 INTRODUCTION

Recently, low-cost and small RGB-D sensors appeared. These sensors can acquire colored 3D models using color image and depth data. The 3D models of target objects can be reconstructed using depth data, so these RGB-D sensors has caused a surge in 3D perception research in the past few years. However, a limitation of the RGB-D sensors is that sunlight interferes with the pattern projecting LED. Therefore, these sensors are not available in outdoor scenes. Fig.1 shows measurement result in outdoor scenes. The black color represents the area where cannot be measured. As shown the bottom of Fig.1, almost all the area cannot be measured in outdoor scenes.

In this research, we developed a novel measurement method for RGB-D sensors in outdoor scene. This method uses several measurement data from multiple viewpoints, and estimates the shape and the sensor poses using Structure from Motion (SfM) and a scale adjustment method. SfM algorithms have a scale ambiguity problem. Then, our method uses scale information obtained from partially acquired area which is near to the viewpoints, and our method optimizes the shapes and poses by the modified bundle adjustment with the scale information. The bundle adjustment minimizes the reprojection error of the features in the acquired images and the depth error between measurement data and estimated data. At last, our method generates dense point cloud using a multi-view stereo algorithm. Our method obtains the correct scale and reconstructs the shape which locates out of measurement range in outdoor environment. In our experiment, we show that our method can measure the range up to 20m away with 700mm accuracy by measuring from several viewpoints in the range of 5m using a RGB-D sensor in outdoor scenes.

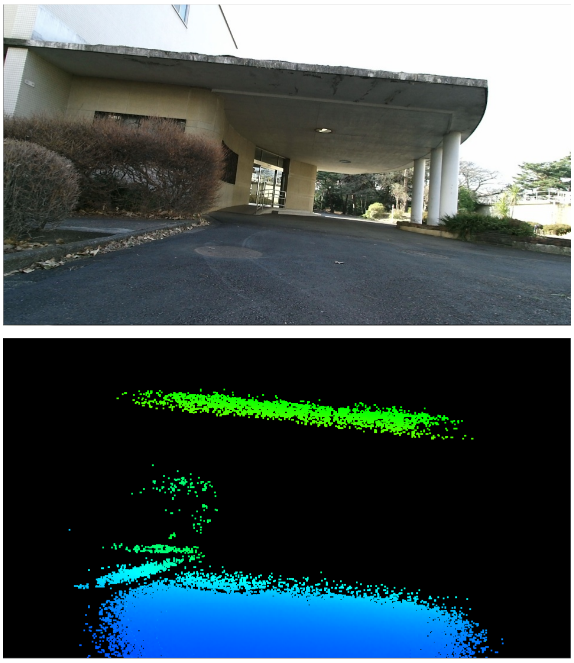


Figure 1: Measurement data from a RGB-D sensor in outdoor scenes. The top shows acquired color image. The bottom shows color mapped depth image. The black colored area represents out of range. The area which is near to the viewpoint is acquired.

2 RELATED WORK

Recently RGB-D sensors have become very popular in the area of Simultaneous Localization and Mapping (SLAM) [Henry10][Endres12]. The major advantage of these sensors is that they provide a rich source of 3D information at relatively low cost. Unfortunately, in outdoor scenes, sunlight affects the measurement result of these sensors, so these sensors are limited to use in indoor scenes.

SfM [Davison07][Snavely07] computes camera poses and 3D shapes of scenes as 3D point cloud using only corresponding feature points in each 2D image. These image-based approaches can be applied to outdoor scenes. To find the correspondences between images, features such as corner points are tracked from one image to another image. However, the image-based method has a scale ambiguity problem [Hartley00]. It is impossible to recover the absolute scale of the scene.

To avoid the scale ambiguity problem, sensor fusion approaches are proposed. Pollefeys et al. [Pollefeys08] integrated captured image sequences with GPS data to correct the scale. Nutzi et al. [Nutzi10] proposed to merge the output of a SfM algorithm with IMU (Inertial Measurement Unit) measurements in an Extended Kalman Filter holding the scale as an additional variable in the state. However, these sensor fusion approaches need additional sensors.

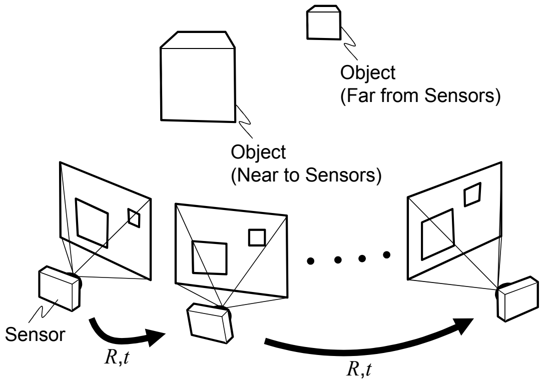


Figure 2: Measurement from several viewpoints. In this case, the near object and the far object exist.

In contrast to these previous works, our method obtains the correct scale and 3D models in outdoor scenes using one RGB-D sensor only without another sensor.

3 PROPOSAL METHOD

Our method estimates the 3D shape of outdoor scenes by a RGB-D sensor using feature points and scale of which distance is acquired. At first, we measure color images and depth images from several viewpoints moving the RGB-D sensor as shown in Fig.2. Next, using the image-based SfM which uses the feature in the acquired images, the 3D shape can be measured robustly in outdoor scenes. However, the correct scale is unknown. Then, we adjust the scale using depth data obtained from acquired area which is near to viewpoints. And our method minimizes the reprojection error and the depth error. The reprojection error is the 2D distance between the features and projected points as shown in Fig.3. The depth error is the distance between the estimated depth and the measured depth as shown in Fig.4. At last, we generate dense point cloud using multi-view stereo.

3.1 Initialization

At first, sensor poses and 3D shapes are initialized using image-based SfM. The SfM computes camera poses and 3D shapes as 3D point cloud using only corresponding feature points in each view. To find the correspondences between images, features such as corner points are tracked from one image to another image. One of the most widely used feature detectors is the SIFT (Scale-invariant feature transform) [Lowe04]. Given a set of corresponding points in two or more images, camera matrices and 3D coordinate of the features are estimated by minimizing reprojection error. In this way, the relative 3D structure of the target scene can be estimated. And we will use the provisional scale which is obtained this initialization step. In our implementation, we used the VisualSFM Software [Wu11].

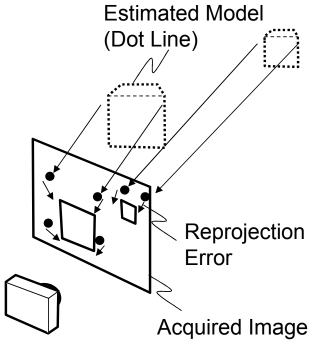


Figure 3: Reprojection Error. It is geometric error corresponding to the 2D image distance between the projected point and the measured point. The error is shown as length of the arrows from projected points.

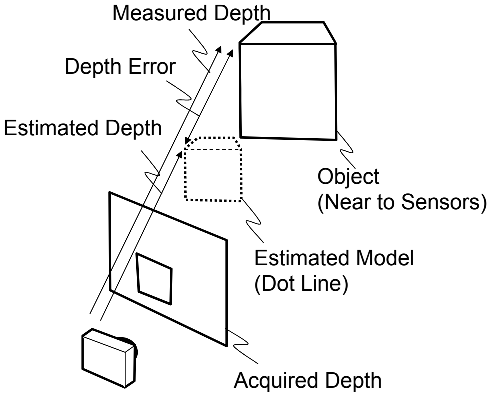


Figure 4: Depth Error. It is a distance between a estimated depth and a measured depth. The estimated depth is obtained by reconstructed shape.

3.2 Scale Adjustment

In this section, we explain the way to adjust scale using depth data obtained from acquired area which is near to viewpoints. Let d_{ij} be the depth of features j in images i, which is obtained by the estimated 3D structure of the target scene with the provisional scale in the above section. Let d'_{ij} be the depth which is obtained from measurement data of feature points. Then the following relation holds

$$d'_{ij} = sd_{ij} \tag{1}$$

where the scale s is the ratio between the provisional estimated scale and the measured scale. However, typically, the above condition does not necessarily satisfy because of the estimation error of SfM and the measurement error of RGB-D sensor. In our method, we compute the optimal scale s^* by minimizing follow equation:

$$s^* = \arg\min_s \sum_i \sum_j (d'_{ij} - sd_{ij})^2. \tag{2}$$

The scale s^* is given by

$$s^* = \frac{\sum_i \sum_j d'_{ij} d_{ij}}{\sum_i \sum_j d_{ij}^2}. \tag{3}$$

Using the scale s^*, the 3D coordinate q of features is updated by

$$q \leftarrow s^* q \tag{4}$$

and the camera pose $\boldsymbol{T}$ is

$$\boldsymbol{T} \leftarrow s^* \boldsymbol{T} \tag{5}$$

where $\boldsymbol{T}$ is 3D vector which represents camera's position.

3.3 Optimization

In this section, we show our optimization approach using a modified bundle adjustment. Conventional bundle adjustments minimize the reprojection error to estimate camera poses and 3D shapes [Triggs00]. In contrast to these bundle adjustment methods, our modified bundle adjustment uses the 3D positions of features and the depth data acquired by RGB-D sensor. In our research, we assume a pinhole camera model. In this model, the mapping from 3D coordinates of points in space to 2D image coordinates can be represented in homogeneous coordinates. Let $\boldsymbol{q}$ be representation of the 3D point in homogeneous coordinates, and let (u', v') be representation of the projected point in the pinhole camera. Then the following relation holds

$$\begin{pmatrix} u'_{ij} \\ v'_{ij} \\ 1 \end{pmatrix} \propto \boldsymbol{P}_i \begin{pmatrix} \boldsymbol{q}_j \\ 1 \end{pmatrix} \tag{6}$$

where $\boldsymbol{P}$ is a 3 × 4 camera matrix which is given by combining a camera calibration matrix $\boldsymbol{K}$, rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{T}$. The camera matrix is given by

$$\boldsymbol{P}_i = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{T}_i \end{bmatrix} \tag{7}$$

where the camera calibration matrix $\boldsymbol{K}$ is an upper triangular matrix that is consist of the focal length and the principal point, the rotation matrix $\boldsymbol{R}$ is a 3 × 3 matrix that represents camera orientation, and the translation vector $\boldsymbol{T}$ is a three vector that represents camera's position. The image points (u', v') given by

$$u'_{ij} = \frac{\boldsymbol{P}_i^{(1)} \begin{bmatrix} \boldsymbol{q}_j^T & 1 \end{bmatrix}^T}{\boldsymbol{P}_i^{(3)} \begin{bmatrix} \boldsymbol{q}_j^T & 1 \end{bmatrix}^T} \tag{8}$$

$$v'_{ij} = \frac{\boldsymbol{P}_i^{(2)} \begin{bmatrix} \boldsymbol{q}_j^T & 1 \end{bmatrix}^T}{\boldsymbol{P}_i^{(3)} \begin{bmatrix} \boldsymbol{q}_j^T & 1 \end{bmatrix}^T} \tag{9}$$

where $\boldsymbol{P}^{(k)}$ is k-th row of the camera matrix $\boldsymbol{P}$. The reprojection error between project points and observed points E_1 is given by

$$\begin{aligned} E_1 &= \frac{1}{2}||\boldsymbol{e}_1||^2 \\ &= \frac{1}{2}\sum_i \sum_j ((u_{ij} - u'_{ij})^2 + (v_{ij} - v'_{ij})^2). \end{aligned} \tag{10}$$

where (u_{ij}, v_{ij}) are the measured feature points, and $\boldsymbol{e}_1$ is residual error. Then, we explain depth error between estimated depth and measured depth. In the pinhole camera model, the depth of features are given by

$$d'_{ij} \;=\; \boldsymbol{P}_i^{(3)} \left[\begin{array}{cc} \boldsymbol{q}_j^T & 1 \end{array} \right]^T . \tag{11}$$

Then, the depth error is given by

$$\begin{aligned} E_2 &= \frac{1}{2}||\boldsymbol{e}_2||^2 \\ &= \frac{1}{2}\sum_i \sum_j (d_{ij} - d'_{ij})^2 \end{aligned} \tag{12}$$

The total error from reprojection error and depth error is given by

$$E = rE_1 + (1-r)E_2 \tag{13}$$

where r is the weight parameter between the 2D distance on the images and the 3D distance in the reconstructed structure. Next, we explain the way to minimize the error. To solve non-linear least squares problems, the Levenberg-Marquardt (LM) algorithm is most widely used. The method interpolates between the Gauss-Newton algorithm and the gradient descent method. This method updates parameters $\boldsymbol{x}$ with

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - (\boldsymbol{H} + \lambda \boldsymbol{I})^{-1}\boldsymbol{g} \tag{14}$$

where $\boldsymbol{I}$ is identity matrix, $\boldsymbol{H}$ is hessian matrix, and $\boldsymbol{g}$ is gradient vector. λ is damping factor which adjusts the step size at each iteration. Using residual error $\boldsymbol{e}_1$ in (10) and $\boldsymbol{e}_2$ in (12), the total error (13) is

$$\begin{aligned} E &= \frac{r}{2}||\boldsymbol{e}_1||^2 + \frac{1-r}{2}||\boldsymbol{e}_2||^2 \\ &= \frac{1}{2}\begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix}^T \begin{pmatrix} r\boldsymbol{I} & 0 \\ 0 & (1-r)\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix} \end{aligned} \tag{15}$$

The gradient vector and the approximated hessian matrix is given by

$$\boldsymbol{g} = \begin{pmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{pmatrix}^T \begin{pmatrix} r\boldsymbol{I} & 0 \\ 0 & (1-r)\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix} \tag{16}$$

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{pmatrix}^T \begin{pmatrix} r\boldsymbol{I} & 0 \\ 0 & (1-r)\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{pmatrix} \tag{17}$$

where the matrix $\boldsymbol{J}$ is jacobian. Using residual error, the jacobian is given by

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{pmatrix} = \begin{pmatrix} \frac{d\boldsymbol{e}_1}{d\boldsymbol{x}} \\ \frac{d\boldsymbol{e}_2}{d\boldsymbol{x}} \end{pmatrix} \tag{18}$$

Note that we gave the approximated hessian matrix. The hessian matrix is a symmetric positive definite matrix and the solution to (14) can be obtained.

Unfortunately, due to the large number of unknowns contributing to the minimized reprojection error and depth error, the computational cost of LM algorithm becomes high. The cost depends on computing cost of inverse of the hessian matrix. To solve this problem, the Sparse Bundle Adjustment (SBA) [Triggs00] takes advantage of the sparse structure of the hessian matrix. The SBA solves huge minimization problems over many thousands of variables within seconds on a standard PC for the image-based SfM. Fortunately, in our method, it is easy to apply sparse technique to deal with both reprojection error and depth error. In the SBA formula, replacing our jacobian matrix and our residual vector to original SBA's jacobian matrix and residual vector, this problem can be solved.

3.4 Generating Dense Point Cloud

Our algorithm uses only feature points, so reconstructed 3D structure is sparse. Then, at last, we apply multi-view stereo algorithm [Seitz06] which generates dense point cloud. The multi-view stereo algorithm uses the camera poses estimated in above section. Using the poses, our method generates dense point cloud whose scale is correct. In our research, we apply The Patch-based Multi-view stereo (PMVS) [Furukawa10] to generate dense point cloud.

4 EXPERIMENTAL RESULT

In this section, we show our experimental result. We test our method using Kinect v2 in outdoor scenes. We compared our result with ground truth data which is acquired by high accuracy Laser Range Finder (LRF). As a LRF, we choose a RIEGL VZ-400. The accuracy of VZ-400 is about 5mm at 100m range.

We measured from 70 viewpoints in the range of 5 meters using RGB-D sensor in outdoor scenes. The top of Fig.1 shows the acquired RGB image, and the bottom of Fig.1 shows the acquired depth map. In the bottom of Fig.1 black color represents the area where cannot be measured. The top of Fig.5 shows the result of the estimated depth map which is estimated by our method, and the bottom of Fig.5 shows the ground truth. As the Fig.5, the resolution of the depth map in our method is lower than the ground truth. The reason is that the number of points generated PMVS is less than LRF data. For visibility, we determined the resolution according to the number of points. As shown in Fig.5, our result can estimate the depth out of RGB-D sensor's range. Fig.6 shows the estimated accuracy. The line depicts the root mean square (RMS) error. RMS is computed using the difference between our result and ground truth around each place. The graph shows our method can estimate the range up to about 20 meters away with 700 mm accuracy. And accuracy is higher in the location close to the sensor. The running time of our scale adjustment and bundle adjustment is 16.5s and 60MB memory is used in this case.

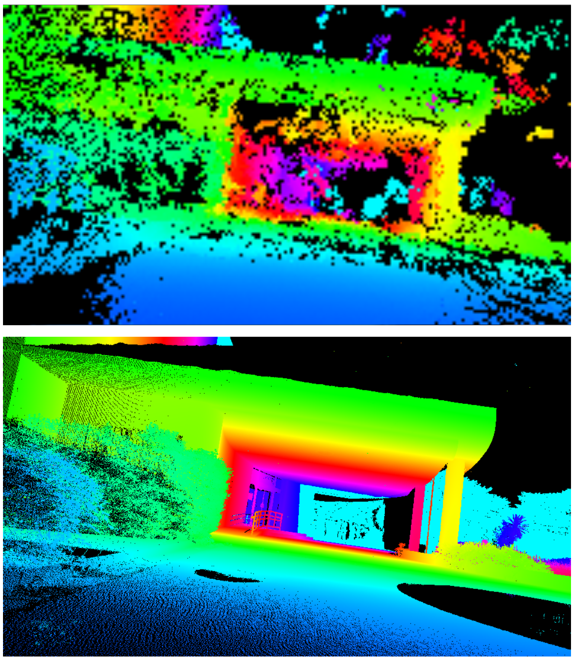


Figure 5: Color mapped depth image. The top shows estimated depth image using our method. The bottom shows ground truth.

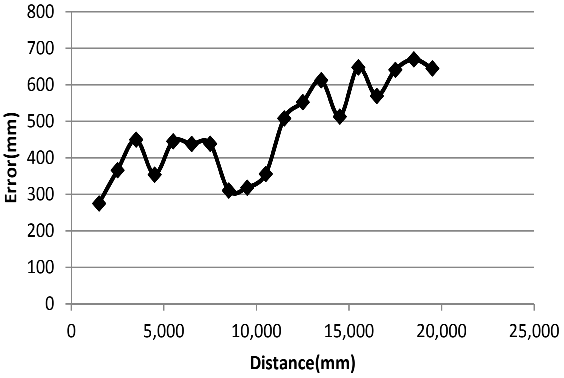


Figure 6: Accuracy of our method. The line shows the root mean square error which is computed using the difference between our estimated depth and ground truth.

Above described scene is measured in the shade, so the acquired image is not bright as shown Fig1. Then, we measured from 90 viewpoints in the range of 5 meters using RGB-D sensor in the direct sunlight (not against sun). The top of Fig.7 shows the acquired RGB image, the middle of Fig.7 shows the generated depth map, and the bottom of Fig.7 shows the ground truth. Fig.6 shows the estimated accuracy. The graph shows our method can estimate the range up to about 20 meters away with 500 mm accuracy. The result shows that our method can work in the bright environment. In this case, the running time of our scale adjustment and bundle adjustment is 69.2s and 135MB memory is used.

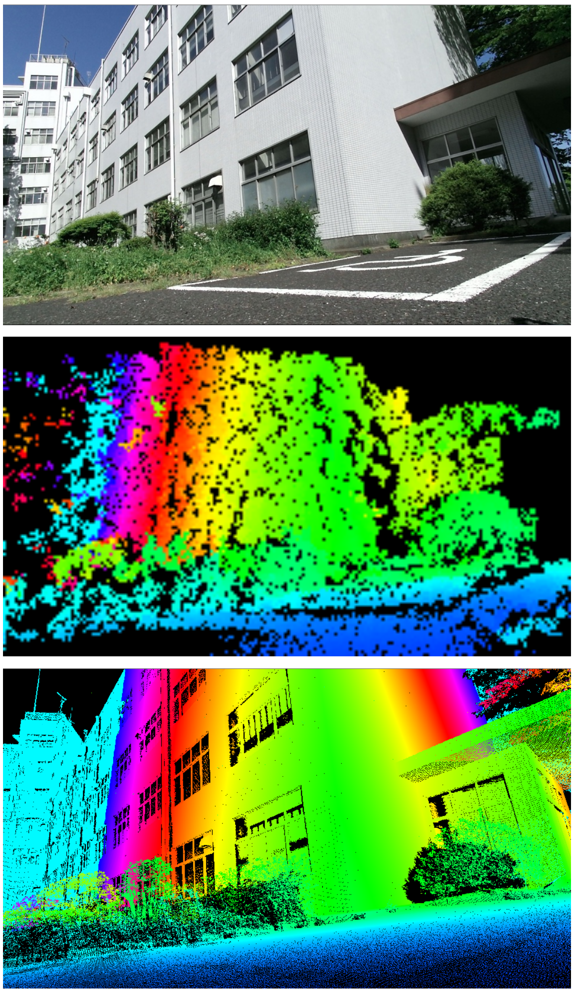


Figure 7: Color mapped depth image in the bright scene. The top shows color image, the middle shows estimated depth image. The bottom shows ground truth.

5 DISCUSSION

As Fig.5 and Fig.7, the result shows the our depth maps became noisy. The reason is correspondence error between color images in the process of MVS described in Sec.3.4. For example, error models are generated in the sky (Fig.5). And our method cannot reconstruct the ground surface model although the ground exists in the captured area. In our method, MVS algorithm generates dense 3D points using correspondence of image patches, so the method cannot make correspondence on texture-less area. Therefore, it is difficult to reconstruct the model of these texture-less areas correctly.

As shown the graph, the accuracy from 4m to 6m became low. In this area, only ground surface exists, so the error became large. The reason of this low accuracy is correspondence error as described above.

Next, we will discuss the accuracy of our result. In our method, we use PMVS algorithm for generating dense point cloud. However, the accuracy of our re-

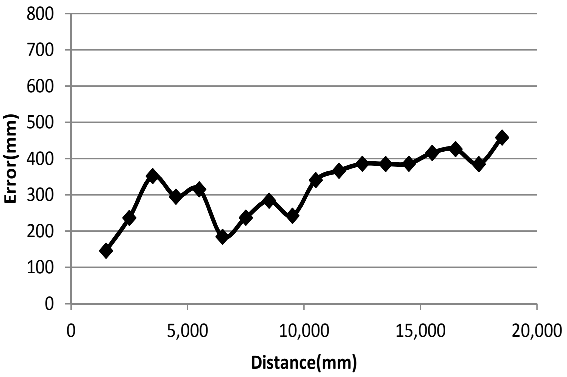


Figure 8: Accuracy of our method in the bright scene.

sults was less quality than the original results of PMVS [Furukawa10]. The accuracy of reconstructed models depends on the accuracy of obtained camera poses in PMVS. In our method, the camera poses are estimated by our bundle adjustment, so the accuracy is chiefly affected by the bundle adjustment. One of the reasons for the low accuracy is that we did not use robust algorithm in our current implementation, in particularly the estimated scale influences the accuracy because the error becomes larger as the object leaves more the sensor position. The scale is determined by the ratio of estimated distance by the features to acquired distance directly.

6 CONCLUSION

We developed a measurement method which reconstructs 3D shapes in outdoor scenes. This method uses measurement data acquired from multiple viewpoints, and estimates the 3D shape and the sensor poses using reprojection error and depth error. Our method obtains the correct scale in the area that could not be measured directly from RGB-D sensor using both the acquired color images and depth images. In our experiment, we show that our method can measure the range up to 20 meters away with 700 mm accuracy by measuring from several viewpoints in the range of 5 meters using RGB-D sensor in outdoor scenes.

As a future work, we plan to apply robust approaches and to compare the accuracy of our method with another approach that can determine the scale. And we will extend to on-line algorithm using local bundle adjustment and video-based real-time MVS.

7 REFERENCES

[Davison07] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse, “MonoSLAM: Real-time Single Camera SLAM” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 29, No. 6, pp. 1052–1067, 2007.

[Endres12] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard, “An Evaluation of the RGB-D SLAM System,” In Proc of the IEEE Int’l Conf. on Robotics and Automation, 2012.

[Furukawa10] Y. Furukawa and J. Ponce, “Accurate, Dense, and Robust Multi-View Stereopsis,” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 32, No. 8, pp. 1362–1376, 2010.

[Hartley00] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision,” Cambridge University Press, 2000.

[Henry10] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments,” in Proc. Int’l Symp. Experimental Robot, 2010.

[Lowe04] D. G. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” Int’l J. of Computer Vision, Vol. 60, No. 2, pp.91–110, 2004.

[Nutzi10] G. Nutzi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM,” Journal of Intelligent Robotic Systems, vol. 61, pp. 287–299, 2010.

[Pollefeys08] M. Pollefeys, D. Nister, J. M. Frahm, A. Akbarzadeh, et al, “Detailed Real-time Urban 3D Reconstruction from Video,” Int’l J. of Computer Vision, Vol. 78, No. 2–3, pp.143–167, 2008.

[Seitz06] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, “A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms,” In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition , Vol. 1, pp. 519–526, 2006.

[Snavely07] N. Snavely, S. Seitz, R. Szeliski, “Modeling the World from Internet Photo Collections,” Int’l J. of Computer Vision, Vol. 80, No. 2, pp. 189–210, 2008.

[Triggs00] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, “Bundle Adjustment - a Modern Synthesis,” LNCS, Springer Verlag, Vol. 1883, pp. 298–375, 2000.

[Wu11] C. Wu, “VisualSFM: A Visual Structure from Motion System,” `http://homes.cs.washington.edu/~ccwu/vsfm`.

Introducing Aesthetics to Software Visualization

David Baum
University of Leipzig
Grimmaische Strasse 12
04109, Leipzig, Germany
david.baum@uni-leipzig.de

Abstract

In software visualization, but also in information visualization in general, there is a great need for evaluation of visualization metaphors. To reduce the amount of empirical studies a computational approach has been applied successfully, e.g., to graph visualization. It is based on measurable aesthetic heuristics that are used to estimate the human perception and the processing of visualizations. This paper lays a foundation for adopting this approach to any field of information visualization by providing a method, the repertory grid technique, to identify aesthetics that are measurable, metaphor-specific, and relevant to the user in a structured and repeatable way. We identified 25 unique aesthetics and revealed that the visual appearance of the investigated visualizations is mainly influenced by the package structure whereby methods are underrepresented. These findings were used to improve existing visualizations.

Keywords
Aesthetics, Repertory Grid Technique, Software Visualization

1 INTRODUCTION

The benefit of a specific visualization depends upon many factors, such as addressed end-user (e.g., developer, project manager, or client), method of representation and its use case. Therefore it is not possible to prove the superiority of one kind of visualization over another one in general, regardless of dimensionality, method of representation and other influencing factors. Every method of representation, usually called metaphor, has to be empirically evaluated on its own [Kos03]. Unfortunately, only a minority of approaches in software visualization has been empirically validated yet [WLR11a, TC08]. This disproportion is caused by the time effort as well as the expense that is necessary for the implementation of empirical studies such as controlled experiments [LBI^{+}12]. Due to many influencing factors a single experiment is often insufficient and therefore a series of experiments is required [IW03]. In practice the resulting overall costs are a problem, because the empirical studies have to be repeated with every slight modification of the metaphor or for example the layout algorithm. Further a controlled experiment can only show if the use of a metaphor is more effective or more efficient than the used object of comparison. Unfortunately, such experiments provide no explanation for measured values and thus no indication on how to improve it.

For these reasons we want to take a different approach for evaluating software visualizations that has the potential to reduce the required number of empirical studies. It is based on a cognitive model which explains the perception and processing of visualizations [Nor04]. This process is influenced by aesthetic heuristics and they in turn depend on the underlying data [BRSG07]. Once we know the relations between these elements we are able to evaluate visualizations by computation instead of empiricism. However this goal can only be achieved in several stages, and this paper is the first one. This is to identify user-relevant aesthetics. Therefore this paper focuses on the following contributions:

(1) We show, how the repertory grid technique can be applied to software visualization for identifying metaphor-specific aesthetics systematically.

(2) We provide a classification scheme to evaluate the influence of the depicted software entities and to draw comparisons between different metaphors.

(3) We present a study that identified aesthetics for the recursive disk metaphor and revealed undesired side effects.

2 RELATED WORK

The concept of aesthetics is widely used in the field of graph visualization especially for evaluating graph layout algorithms [Hua14, BRSG07]. Basic research was done by Purchase [Pur97, PMCC01, PAC02, PCA02] which led to increased research activity of several authors [BRSG07, WPCM02]. All in all at least 18 graph aesthetics were proposed by different researchers although only a few of them were evaluated empirically [BRSG07]. Due to its successful use the concept of aesthetics has already been transferred to other domains such as UML class and use case diagrams [PMCC01]. However none of these approaches includes a method for identifying aesthetics that are relevant to the user. All of the proposed aesthetics, e.g., in graph visualization have been found without user involvement. This might be a reason why only a minority of the proposed aesthetics has a significant effect with respect to the user performance.

We are not aware of any application in the field of software visualization. In addition no other approach, which has the goal to evaluate visualizations by computation does exist so far.

3 SOFTWARE VISUALIZATION

In software visualization the structure, behavior and/or evolution of software systems are visualized to facilitate a better understanding of abstract and complex software artifacts. Over the last years several visualization metaphors were proposed such as the recursive disk metaphor [MZ15] and the city metaphor [WLR11b]. They differ among other properties in the depicted information and in the used glyphs for representing software entities. *"A glyph is a small visual object that can be used independently and constructively to depict attributes of a data record or the composition of a set of data records"* [BKC$^+$13]. Glyphs use *visual channels* such as shape, color and size to depict information [BKC$^+$13] and they can contain other glyphs as well [War02].

Within this paper we want to focus on the recursive disk metaphor. It uses a glyph-based approach to visualize the structure of software systems. Each software entity, i.e., attributes, methods, classes, and packages is mapped to a different glyph [MZ15]. An attribute is represented by a yellow ring segment. Method glyphs are blue ring segments and their size corresponds roughly to the methods number of statements (NOS). Classes and packages are depicted by purple and gray rings respectively any they may contain several attribute, method and class glyphs. With these basic glyphs even large structures can be visualized. Figure 1 shows a package that contains some exemplary classes. For a more complex real world example see Figure 2. It visualizes the structure of *JUnit 4*, one of the most frequent used test frameworks for Java [JUn15].

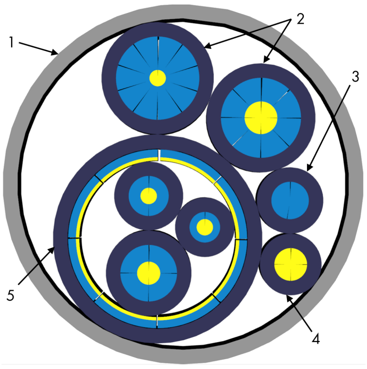


Figure 1: Basic glyphs and relations with the recursive disk metaphor: 1 - Package with five classes, 2 - General classes with altogether eighteen methods and five attributes, 3 - Method class with two methods, 4 - Data class with four attributes, 5 - Class with eight methods, eight attributes, and three inner classes [MZ15].

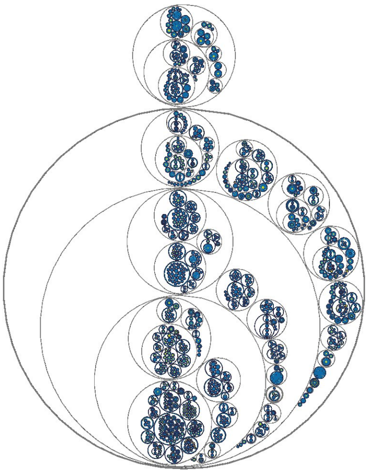


Figure 2: The structure of JUnit 4.12 visualized with the recursive disk metaphor.

4 AESTHETICS

Since aesthetics originates in graph visualization we want to introduce them using a corresponding example (c.f. Figure 3). All three node-link diagrams visualize the same graph and only differ in their layout algorithm. Readability as well as understandability is decreased from left to right. This is caused by their different appearance and not only detectable through empirical experiments but also directly measurable. For example the diagrams differ among other things in their number of edge crossings and in their degree of symmetry. which are both basic aesthetics. Studies haven shown, that reducing edge crossings and increasing symmetry will lead to a faster perception and a better understanding of graphs. If a further layout algorithm will be evaluated with respect to understandability and readability no further empirical studies are necessary. By taking the number of edge crossings and the degree of symmetry into account we can predict the mentioned properties of the graph.

Since a unified definition for aesthetics is missing in the literature, we define them as follows: *Aesthetics are visual properties of a visualization that are observable for human readers as well as directly measurable.* By this definition abstract attributes such as complexity are not considered as aesthetics. Indeed the complexity of a visualization can be measured as well, but only after it was operationalized.

Aesthetics can be used in two different ways: On the one hand certain aesthetics can be optimized, so that the resulting graph is easier to understand for human readers [BRSG07]. This can be measured by the time needed by a user to solve different tasks and the number of errors he made while doing so [Hua14]. On the other hand due to the different appearance of the three node-link diagrams a user would expect the graph on the right side to be more complex than the graph on the left side [PAC02]. It is a basic purpose of a model that assumptions about its original are made. If the visualizations of two systems differ clearly in one point the beholder assumes that this is caused by the underlying data. But in some cases these differences are merely undesired side effects without any reasonable meaning. The resulting assumptions would be at best useless or even wrong as in the mentioned example since all graphs share the same degree of complexity.

However, the use of aesthetics in software visualization, especially when the third dimension is used, is hindered by the lack of methods to identify new aesthetics in a structured and repeatable way. If 18 aesthetics would be proposed for every visualization metaphor, the need for empirical studies is not reduced but increased. This is the key problem that has to be solved in order to apply aesthetics efficiently to a wide field of applications. Therefore we used the repertory grid technique to identify metaphor-specific aesthetics as described in the following section.

5 RESEARCH DESIGN

For the identification of user-relevant and metaphor-specific aesthetics empirical studies are still required. Therefore we investigated the perception of different software systems focusing on the following research questions:

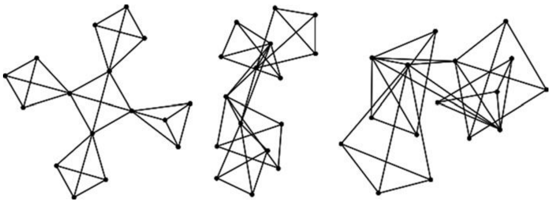


Figure 3: Three node-link diagrams of the same graph [Hua14]

(1) Which aesthetics affect the perception of a software visualization based on the recursive disk metaphor?

(2) How strong is the influence of different software entities on the perception of a software visualization?

Hence the study is based on an exploratory research design. Since the method for identifying relevant aesthetics is the main contribution of this publication the methodology of the applied repertory grid method is described in general as well as our implementation below in detail.

5.1 The Repertory Grid Technique

The repertory grid technique is an empirical research method that originates in psychology. It was already adopted to other research fields such as marketing and software engineering, e.g., to analyze soft skills of software engineers [KSB06], to identify aspects in requirements engineering and risks in software development projects [EMM09] and to evaluate the user experience [VLR$^+$10]. As you will see below neither the method in general nor our implementation are specific to software visualization but also applicable to any kind of information visualization.

The theoretical foundation of the repertory grid technique is the “theory of personal constructs“ developed by G. Kelly [Kel55]. Its basic assumption is that everybody constructs its own reality through his individual perception of the world. To describe and evaluate elements of this world as well as to distinguish between them, people use *bipolar constructs*. A construct is defined as “a way in which two or more things are alike and thereby different from a third or more things“ [Kel55, p. 61]. These construct consist of a construct pole (e.g. “reliable“), a contrast pole (e.g. “unreliable”) and a construct continuum in between, i.e., different degrees of reliability. For example, regarding user experience a user categorizes software *A* as fast, reliable and ugly and software *B* as slow, unreliable and handsome. Thereby he used three bipolar constructs, “fast – slow“, “reliable – unreliable“ and “ugly – handsome“ which means these are the relevant attributes for him in which the two systems differ. The repertory grid method is an approach to make these constructs explicit and visible. Within the terminology of the repertory grid technique software *A* and *B* are referred to as *elements*, that are described through *constructs* [Fra04, p. 15].

A repertory grid interview consists of several steps. For each step multiple design decisions have to be made, e.g., about how elements and constructs are selected. In the following we do not discuss all possibilities, but the research design that corresponds to our research questions, hence variants such as constructs provided by the researcher are not reasonable for exploratory research. In the first step the investigated elements have to be chosen through the participant or the researcher. In our case visualizations of software systems were used as elements. Second, these elements are combined into triads. This can be done randomly or by creating combinations of particular interest. Then the triads are presented successively to the participant, i.e., the participant sees exactly three elements at the same time without access to the other elements.

For each triad presented, the participant has to answer the following question: “How are any two of these alike in some way?“, complemented by “What is the opposite of that?“ [Fra04, p. 29]. The answer to the first question is the construct pole and the answer to the second is the contrast pole. It might happen that these abstract constructs lead to further constructs if they are investigated in depth. It is not uncommon that a construct implies another construct. They only vary in their level of abstraction. The process of using a construct to attain a construct on a different level of abstraction is called *laddering* and is a common part of the repertory grid interview [Fra04, p. 39]. This can be done by asking “Why appears this software more reliable to you?“. The whole procedure is repeated by using other randomly selected triads as long as the participant creates new constructs to distinguish between the elements. During the interview the participants have no access to any constructs they used before. It is crucial that the interviewer understands what exactly the participant describes with a construct. For this reason informal communication between both persons is a regular and intended part of the repertory grid technique.

Once no new constructs can be elicited, the participant has to assess all elements in consideration of all mentioned constructs. For twelve elements and, e.g., 15 constructs the participant has to make 180 decisions. Since a repertory grid interview is already exhausting for the participant often a 5-point rating scale is used to simplify the process [EMM09]. During this phase the participant has access to all 12 visualizations at any time.

To sum up, the participants assess specified elements on constructs they create. As the repertory grid technique demands high standards of the interviewer and of the research design, a pretest is advisable to detect unexpected difficulties.

Once the interviews are conducted successfully the resulting grids can be analyzed in multiple ways. For a qualitative interpretation the results are visualized as a two-dimensional space based on a principal component analysis (PCA) [Fra04, p. 86]. The distances between elements in this space represents their similarity. Small distances represent a high degree of similarity, whereas large distances indicate a low degree of similarity.

Given that most constructs are mentioned only by one or two participants a quantitative analysis of the constructs is not directly possible. Therefore it is necessary to abstract from concrete constructs. This is done by assigning all constructs to given categories [Fra04, p. 49], e.g., “system behavior“ and “appearance“ in the case of the example mentioned before. This way it is possible to draw comparisons between grids of different participants, even if these grids are based on different elements and the participants used different constructs.

5.2 Study Design

The repertory grid technique is a very flexible method, thus it can be applied in many ways. In the following we describe our conducted research design and explain the design decisions we made.

Twelve visualizations with the recursive disk metaphor were used as elements. They were preselected based on static code metrics such as NOS and their number of packages, classes, methods and attributes. Because the recursive disk metaphor visualizes the structure of a system the metrics were limited to structural aspects as well. Except for NOS these structural entities are directly represented in the visualization by glyphs [MZ15]. The NOS were included since they are used to ensure that small, medium and large systems are used in the study. Table 1 gives an overview over the chosen elements. In addition the visualizations can be accessed online [1]. The triads were created randomly and individually for each interview.

The InstantPlayer which is part of the InstantReality platform [fCGRI15] was used to show the visualizations. The participants could freely navigate within the model, i.e., changing the viewpoint along the x-, y- and z-axis which includes zooming and changing the field of view. Each visualization was presented on a separate but similar screen.

We conducted eight repertory grid interviews with four male and four female participants. Their age varies between 19 and 52 years. None of them has seen a recursive disk visualization before. To ensure the participants focus on the visual differences and not on the underlying structural differences of the software, we have not explained the meaning of the shown visualizations to them. The visualizations were only referred to as “model 1“ etc., so they did not even know that the study was about software visualization. We just determined how the participant should call the glyphs to guarantee interviewer and participant talk about the same. This was necessary because in the pretest we detected

[1] `https://github.com/naraesk/aesthetics`

that every participant uses different terms for naming the same glyph. We chose easy-to-understand names, therefore the wording differs from the regular vocabulary of the recursive disk metaphor: The glyphs were called “gray rings“, “purple disks”, “blue segments“ and “yellow segments“. Furthermore the terms “outer gray rings“ and “inner gray rings“ were introduced. The first one describes the root package disks and the second term names the remaining package disks. In addition “element“ was defined to describe any glyph regardless of its shape or color. Besides these small adjustments, we could apply repertory grid, as described in section 5.1, without modification.

The next step after conducting the interviews is to categorize the identified constructs. The means of construct categorization were proposed by Landfield [Lan71] and especially for psychological grids there exist different categorization schemes. Since we applied the repertory grid technique to a new field, existing categorization schemes are insufficient. Therefore we used a simple scheme for object-oriented software systems based on the entities system, package, class, method, and attribute. Every construct is now mapped to exactly one category depending on which entity is the object of comparison. We explain the categorization process using the construct “many purple disks – few purple disks“ which refers to the number of classes. Two packages can differ in their number of classes, for a class itself this is not possible. Therefore the category of the construct is *package*. Of course systems can differ in their number of classes as well, but only the least entity is used. Otherwise *system* would be used for every single construct.

6 RESULTS AND DISCUSSION

The goal of this study was to reveal aesthetics for the recursive disk metaphor. Therefore we refrain from doing a large qualitative statistical interpretation of the individual constructs, because once the aesthetics were extracted it is more meaningful to examine them in a separate study. In this spirit the described study can be seen as a data collection to prepare a more comprehensive quantitative study. Still, we present an overview of the identified constructs. The complete raw data is provided online as well [1].

The study revealed 53 unique constructs that were mentioned by the participants although not all of them are aesthetics since not all of them can be directly measured. Table 2 shows all mentioned constructs and their frequency, which we will discuss briefly. To determine the frequency we had to decide whether two constructs have the same meaning or not, which is not trivial. The measurable constructs tend to be precise and for the participants they were easy to explain by pointing on a concrete example on the screen. A construct such as “number of gray disks“ it is easy to understand and therefore we were able to count these constructs. In case of less precise terms this procedure is problematic, because participants often use a slightly different wording. Therefore we only counted identical constructs. Some of the rarely used constructs may have the same meaning, but still appear as two independent entries, e.g, “chaotic – logical“ (#10) and “regular – irregular“ (#17) could be considered equal. On the one hand Table 2 shows that eleven identified constructs were used by at least half of the participants. On the other hand 26 constructs were mentioned only once. Actually the aesthetics that were mentioned more often were used more often within an interview too. However, for example the construct “dynamic - static“ was mentioned only by one participant nevertheless it seems to be important to him since he used it several times. That just shows that it is not reasonable to focus only on the frequent aesthetics but rather one must take all of them into account. Due to the fact that over 50% immeasurable constructs were used by the participants it seems that in some cases the laddering was not done intensively enough. Nevertheless, this only leads to less data but not false data and does not affect the identified aesthetics.

6.1 Categorization

The categorization revealed that the recursive disk metaphor tends to emphasize packages. Three of the four most frequent aesthetics refer to package disks as objects of comparisons, 48% overall. Further the package structure affects 60% of the identified aesthetics. This means that the visual appearance as well as the assumptions about the underlying data is primarily influenced by how classes are grouped into packages. From our point of view this is neither intended nor desirable. In program understanding as well as in program analysis methods are the center of interest. For instance most pattern as well as anti-pattern are detected on method level [GHJV93, Lan06]. Whereas not a single aesthetic was identified, that refers to methods. Merely three aesthetics use classes as objects of comparisons. In case an aesthetic depends on the class structure it only focuses on the absence of methods (#51) or inner classes (#42). Although the metaphor provides more information such as the size of individual methods, the number of methods of a class and the proportions of these glyphs. But none of these aspects of the visualization were recognized by the participants.

The last column of Table 2 shows which influencing factors an aesthetic has, whereby *everything* means, that this aesthetic is influenced by the number of the different software entities, NOS and the layout algorithm. This is, e.g., the case for the density of a package. The density specifies how much of the area of a disk is filled. By a density of 100% there is no empty space inside

	Packages	Classes	Methods	Attributes	Statements
android_packages_apps_Phone	3	262	1,426	1,394	15,113
android_packages_apps_Settings	40	1,351	5,516	6,140	51,489
apache Storm	71	3,203	2,129	7,417	35,295
ChatSecureAndroid	32	620	2,257	3,334	22,371
cw-omnibus	339	1,224	2,545	7,212	36,696
disruptor	16	271	696	1,100	4,274
FreeFlow	17	73	242	467	2,014
libsvm	2	34	136	134	2,333
JUnit	66	1,163	773	3,704	10,736
Roboguice	98	1,039	2,574	7,014	30,317
Tachyon	18	913	2,153	7,255	43,808
ua-parser	5	35	58	123	741

Table 1: Code metrics of the selected elements

the disks. Changes regarding the number of classes, the NOS of methods or the placement strategy for the glyphs affects the density of the visualization. Some aesthetics are influenced by multiple factors and such a factor in turn affects multiple aesthetics at once. That is why we were not able to extract meaningful factors via a PCA based on the object of comparison or the cause. The explained variance was always insufficient – below 50% for most factors. To obtain more satisfying results about one factor per aesthetic would be necessary. All in all we can say that the relations between aesthetics and their cause in the underlying data are quite complex.

6.2 Implications for the Recursive Disk Metaphor

Based on the results of the study, i.e., identified aesthetics and applied categories we want to identify some drawbacks of the recursive disk metaphor. Further we suggest some modifications to improve the metaphor. The identified aesthetics indicate that it will not be possible to detect typical anti-pattern such as brain methods, brain classes or god classes although the metaphor was expected to be useful for this task [MZ15]. Therefore as a first direct consequence of this study we suggest to make methods more visible and distinguishable. Since the participants were able to perceive the size of class and package disks as well as their proportions methods should be visualized in the same way. Furthermore a method can be considered as the smallest unit of a software system. The presentation is more consistent if all structure units share a similar glyph shape. Figure 4 shows a possible modified version of the recursive disk metaphor. Methods are now represented as own disks just as classes and packages and thereby their importance to the appearance is increased. Although this is less space-efficient we prefer it since it leads to a better perception of the visualization.

Multiple participants chose the thickness of the border of package disks as a construct (#5). Currently the width of a border is fixed, but when packages exist, that only contain exactly one other package the two gray borders looks like one thick one as shown in the left of Figure 5. That structure can be considered as common for java source code and it is not apparent how this information can be used to make meaningful conclusions about the software. For example if three nested packages *org*, *apache* and *common* exist it appears as one package disk with a border three times as thick (given that the packages *org* and *apache* do not contain any other elements). To eliminate this undesired visual difference the three borders should be merged into one disk representing the three packages at once as shown in the right of Figure 5.

Another notable construct is the number of outer gray rings (#3), i.e., package disks. The variance is caused by the number of root packages in the source code. Once again this information is not suitable to make

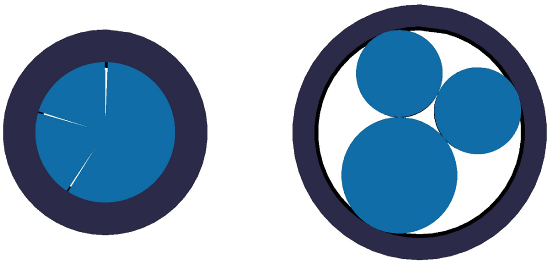


Figure 4: Old (l.) and new (r.) representation of a class with three methods

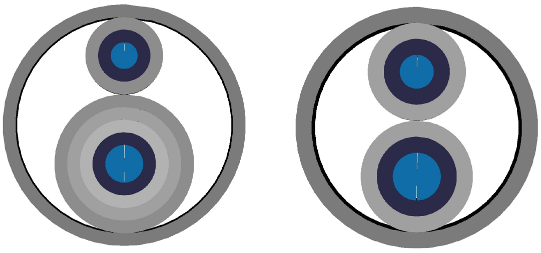


Figure 5: Old (l.) and new (r.) representation of nested packages

statements about the software. If a program has multiple root packages it is always possible to create a new package containing the former root packages. This changes the visual appearance significantly although nearly all code metrics will be unchanged. Therefore we suggest that the visualization should always contain one all-encompassing package disk to unify the appearance regardless of the existence of a corresponding package in the source code.

Furthermore it is conspicuous, that only two participants consider size (#22) as a suitable construct although it seems like a very basic attribute of a visualization. The area of the largest visualization is about 146 as big as the smallest one. This is a sign for a missing scale which makes it nearly impossible to estimate size of a visualization. The user interface did not contain any possibility to compare the size of the displayed objects. However this can not be seen as a disadvantage of the visualization metaphor, because the user interface has to provide these features. Hence it should be modified to provide a visible scale, coordination system or something similar to empower the user to compare the size of different visualizations.

7 FUTURE WORK

The insights provided in this paper lay a foundation for additional avenues for future work on computational evaluation in software visualization. First, a quantified study is in preparation to investigate the cause and the effect of the identified aesthetics. This will help to understand the relations between the underlying data and these aesthetics as well as the extracted factors of the PCAs of the repertory grid interviews. Second, the suggested modifications to the recursive disk metaphor have to be evaluated empirically. Furthermore the repertory grid technique will be applied to more visualization metaphors such as the city metaphor. Thereby we will be able to compare the impact of the depicted software entities between the different metaphors and investigate how this influences the drawn conclusions from a visualization.

8 CONCLUSION

In this paper, we introduced aesthetics to pave the way towards a computational approach of evaluating software visualizations. Although the presented work has only made a first step towards this goal it already provides new means to improve existing visualization metaphors. With the mean of the repertory grid technique we showed how aesthetics can be identified methodically. Through categorizing the found constructs the impact of the different software entities was made comparable. The recursive disk metaphor overemphasizes the package structure and hinders comparisons between methods due to their similar appearance. We suggested some modifications to the glyph shape and placement strategy to counteract.

9 ACKNOWLEDGMENTS

We are grateful to the volunteers who participated in this study.

10 REFERENCES

[BKC+13] R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. O. Ward, and M. Chen. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. 2013.

[BRSG07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The aesthetics of graph visualization. In *Proc. of the Third Eurographics Conf. on Comp. Aesthetics in Graphics, Vis. and Imaging*, Computational Aesthetics'07, pages 57–64, Aire-la-Ville, Switzerland, Switzerland, 2007.

[EMM09] Helen M. Edwards, Sharon McDonald, and S. Michelle Young. The repertory grid technique: Its place in empirical software engineering research. *Inf. Softw. Technol.*, 51(4):785–798, April 2009.

[fCGRI15] The Fraunhofer Institute for Computer Graphics Research IGD. InstantReality. http://www.instantreality.org/, 2015.

[Fra04] Fay Fransella. *A manual for repertory grid technique* . Wiley, Chichester, 2. edition, 2004.

[GHJV93] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Abstraction and Reuse of Object-Oriented Design. *Lect. Notes Comput. Sci.*, 707:406–431, 1993.

[Hua14] Weidong Huang. Evaluating Overall Quality of Graph Visualizations Indirectly and Directly. In Weidong Huang, editor, *Handb. Hum. Centric Vis.* Springer, New York, 2014.

[IW03] Pourang Irani and Colin Ware. Diagramming information structures using 3D perceptual primitives. *ACM Trans. Comput. Interact.*, 10(1):1–19, 2003.

[JUn15] JUnit 4. JUnit 4, May 2015. `https://github.com/junit-team/junit`. Accessed: 2015-05-20.

[Kel55] George A. Kelly. *The psychology of personal constructs. Volume I.* Norton, New York, 1955.

[Kos03] Rainer Koschke. Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey. *J. Softw. Maint. Evol.*, 15(October 2002):87–109, 2003.

[KSB06] V.A. Khamisani, M.S. Siddiqui, and M.Y. Bawany. Analyzing soft skills of software engineers using repertory grid. In *Multitopic Conf., 2006. INMIC '06. IEEE*, pages 259–264, Dec 2006.

[Lan71] A. W. Landfield. *Personal construct systems in psychotherapy*. Rand McNally, Chicago, 1971.

[Lan06] Michele Lanza. *Object oriented metrics in practice : Using software metrics to characterize, evaluate, and improve the design of object-oriented systems*. Springer, Berlin, 2006.

[LBI$^+$12] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *Vis. and Comp. Graphics, IEEE Transactions on*, 18(9):1520–1536, Sept 2012.

[MZ15] Richard Müller and Dirk Zeckzer. The Recursive Disc Metaphor: A Glyph-based Approach for Software Visualization. *6th Int. Conf. on Inf. Vis. Theory and Applications*, 2015.

[Nor04] Donald Norman. *Emotional Design. Why We Love (or Hate) Everyday Things*. Basic Books, 2004.

[PAC02] Helen C. Purchase, Jo-Anne Allder, and David Carrington. Graph Layout Aesthetics in UML Diagrams: User Preferences. *J. Graph Algorithms Appl.*, 6(3):255–279, 2002.

[PCA02] Helen C. Purchase, David Carrington, and Jo-anne Allder. Empirical Evaluation of Aesthetics-based Graph Layout. *Empir. Softw. Eng.*, 7:233–255, 2002.

[PMCC01] Helen C. Purchase, Matthew McGill, Linda Colpoys, and David Carrington. Graph drawing aesthetics and the comprehension of uml class diagrams: An empirical study. In *Proc. of the 2001 Asia-Pacific Symp. on Inf. Vis. - Volume 9*, APVis '01, pages 129–137, Darlinghurst, Australia, Australia, 2001. Australian Computer Society, Inc.

[Pur97] Helen C. Purchase. Which Aesthetic Has the Greatest Effect on Human Understanding? In *Proc. 5th Int. Symp. Graph Draw.*, 1997.

[TC08] Alfredo R Teyseyre and Marcelo R Campo. An overview of 3D software visualization. *IEEE Trans. Vis. Comput. Graph.*, 15(1):87–105, 2008.

[VLR$^+$10] Arnold P O S Vermeeren, Effie Lai-chong Law, Virpi Roto, Marianna Obrist, Jettie Hoonhout, and Kaisa Väänänen-Vainio-Mattila. User experience evaluation methods. *Proc. 6th Nord. Conf. Human-Computer Interact. Extending Boundaries - Nord. '10*, page 521, 2010.

[War02] Matthew O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, December 2002.

[WLR11a] Richard Wettel, Michele Lanza, and Romain Robbes. Software systems as cities: a controlled experiment. *2011 33rd Int. Conf. Softw. Eng.*, pages 551–560, 2011.

[WLR11b] Richard Wettel, Michele Lanza, and Romain Robbes. Software systems as cities: A controlled experiment. In *Proc. of the 33rd Int. Conf. on Soft. Eng.*, ICSE '11, pages 551–560, New York, USA, 2011. ACM.

[WPCM02] Colin Ware, Helen C. Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Inf. Vis.*, 1(2):103–110, June 2002.

#	Construct	Freq	Object of Comparison	Cause
1	low density – high density	8	Package	*everything*
2	heavily nested – little nested	8	Package	package structure
3	many outer gray rings – few outer gray rings	8	System	no. of packages package structure
4	simple – complex	7		
5	thin edge of outer ring – thick edge of outer ring	5	Package	package structure
6	high yellow share – less yellow share	5	Class	no. of attributes
7	centered – off-center	5		
8	elements distributed equally/leaning to the right	5	System	*everything*
9	short strings of purple disks – long strings of purple disks	4	Package	package structure no. of classes
10	chaotic – logical	4		
11	many purple disks – few purple disks	4	Package	no. of classes
12	few/many inner outer rings with the same size	3	System	package structure package size
13	open strings – closed strings	3		
14	many/few outer rings with the same size	3	System	package structure package size
15	many/few purple disks with the same size	3	Package	class size
16	detailed – not detailed	2		
17	many/few gray rings	2	System	package structure
18	harmonic – inharmonic	2		
19	many/few inner gray rings	2	Package	package structure
20	circular/semicircular positioning of elements	2		
21	symmetrical – asymmetric	2	System	*everything*
22	small – large	2		
23	nested gray rings – no nested gray rings	2	System	package structure
24	boring – interesting	2		
25	one layer of purple disks – multiple layers of purple disks	2	Package	no. of classes
26	unstructured - structured	2		
27	predominant simple/complex purple disks	2	Package	class structure
28	manageable – overloaded	1		
29	balanced – not balanced	1		
30	many/few purple disks with a similar structure	1	Package	class structure
31	dynamic – static	1		
32	(no) constantly increasing element size within a gray ring	1		
33	few/many elements in the outer rings	1	System	package structure no. of classes
34	compact – not compact	1		
35	regular – irregular	1		
36	few/many inner gray rings with the same size	1	System	package structure package size
37	beautiful – less beautiful	1		
38	isolated purple disks – no isolated purple disks	1		
39	centered y-axis – shifted y-axis	1	System	*everything*
40	closed – open	1		
41	area with dominant color – no area with dominant color	1		
42	yellow rings – yellow disks	1	Class	class structure
43	orderly – disordered	1		
44	simple spiral shape – complex spiral shape	1		
45	small purple disks – large purple disks	1	Class	class size
46	complicated – uncomplicated	1		
47	loose structure – compact structure	1		
48	yellow evenly distributed/concentrated on one area	1	Package	class structure
49	coarse structure – fine structure	1		
50	flat – deep	1		
51	few/many purple disks without blue segments	1	Package	class structure
52	homogeneous/heterogeneous inner gray rings	1		
53	artificial – natural	1		

Table 2: List of identified constructs. Aesthetics are highlighted gray.

Hyperspectral Image Classification Using a General NFLE Transformation with Kernelization and Fuzzification

Ying-Nong Chen	Yu-Chen Wang	Chin-Chuan Han	Kuo-Chin Fan
National Central University Department of Computer Science and Information Engineering Taiwan, Taoyuan City	National Central University Department of Computer Science and Information Engineering Taiwan, Taoyuan City	National United University Department of Computer Science and Information Engineering Taiwan, Miaoli City	National Central University Department of Computer Science and Information Engineering Taiwan, Taoyuan City
yingnong1218@gmail.com	m09502062@chu.edu.tw	cchan@csie.ncu.edu.tw	kcfan@csie.ncu.edu.tw

ABSTRACT

Nearest feature line (NFL) embedding (NFLE) is an eigenspace transformation algorithm based on the NFL strategy. Based on this strategy, the NFLE algorithm generates a low dimensional space in which the local structures of samples in the original high dimensional space are preserved. Though NFLE has successfully demonstrated its discriminative capability, the non-linear manifold structure cannot be structured more efficiently by linear scatters using the linear NFLE method. To address this, a general NFLE transformation, called fuzzy/kernel NFLE, is proposed for feature extraction in which kernelization and fuzzification are simultaneously considered. In the proposed scheme, samples are projected into a kernel space and assigned larger weights based on that of their neighbors according to their neighbors. In that way, not only is the non-linear manifold structure preserved, but also are the discriminative powers of classifiers increased. The proposed method is compared with various state-of-the-art methods to evaluate the performance by several benchmark data sets. From the experimental results, the proposed FKNFLE outperformed the other, more conventional, methods.

Keywords

Hyperspectral image classification, manifold learning, nearest feature line embedding, kernelization, fuzzification

1. INTRODUCTION

Dimensionality reduction (DR) in hyperspectral image (HSI) classification is a critical issue during data analysis because most multispectral, hyperspectral, and ultraspectral images generate high-dimensional spectral images with abundant spectral bands and data. However, it is challenging to classify these spectral data because vast amount of samples have to be collected for training beforehand. Besides, the spectral properties of land covers are too similar to clearly separate them out. Hence, an effective DR is an essential step to extract the salient features for classification. Recently, a number of DR methods have been proposed that can be classified into three categories: linear analysis, manifold learning, and kernelization. Those using linear analysis try to model the linear variation of samples and find a transformation to maximize the global scatter matrix, e.g. principal component analysis (PCA), linear discriminant analysis (LDA), and discriminant common vectors (DCV). Sample scatters are represented in the global Euclidean structure in these methods. They work well for DR or classification if samples are linearly separated or are distributed in a Gaussian function. However, when samples are distributed in a manifold structure, the local structure of a sample in a high dimensional space is not apparent when using global measurement. In addition, the classification performance in the case of linear analysis methods would deteriorate when the decision boundaries are predominantly nonlinear. Manifold learning methods are proposed to reveal the local structure of samples. He et al. propose the locality preserving projection (LPP) method to preserve the local structure of training samples for face recognition. Since LPP presents sample scatter using the relationship between neighbors, the local manifold structure is preserved and the performance is more effective than in the case of the linear analysis methods. Similar to LPP, Tu et al. propose a Laplacian eigenmap (LE) method for land cover classification using polarimetric synthetic aperture radar data. The LE algorithm reduces the dimensions of features from a high-dimensional polarimetric manifold space to an intrinsic low-dimensional manifold space. Wang and He investigated the LPP for DR in HSI classification. Kim et al. utilized the locally linear embedding (LLE) method to reduce the dimensionality of HSIs. Li et al. propose the local Fisher discriminant analysis (LFDA) method which integrates the properties of LDA and LPP to reduce the dimensionality of HSI data. Luo et al. propose a discriminative and supervised neighborhood preserving embedding (NPE) method for feature extraction in HSI classification. These manifold learning methods all preserve the local structure of samples and improve on the performance of conventional linear analysis methods. However, the applicability of linear manifold learning is limited to noises. Generally, the discriminative salient features of training samples are extracted using certain evaluation processes. An appropriate kernel function could improve the performance for the given method[13]. The kernelization approaches have been proposed for improving the performance of HSI classification. Boots and Gordon introduced a kernelization method to alleviate the limitation of manifold learning. Scholkopf et al. propose a kernel PCA (KPCA) method for nonlinear DR. KPCA generates a high-dimensional Hilbert space to extract the non-linear structure that is missed by PCA. Furthermore, Lin et al. propose a general framework for multiple kernel learning during DR. They unify the multiple kernel representation, and the multiple feature representations of data are consequently revealed in a low dimension. On the other hand, a composite kernel scheme, a linear combination of multiple kernels, extracts both spectral and spatial data. Chen et al. present a sparse representation of kernels for HSI classification. A query sample is represented via all training samples in an induced kernel space. Moreover, pixels within a local neighborhood are also represented by the combination of training samples. In the previous works, the nearest feature line (NFL) strategy was embedded into the linear transformation for dimension reduction on face recognition and HSI classification. However, the nonlinear and non-Euclidean structures are not efficiently extracted using the linear transformation. Fuzzification and kernelization are two efficient tools for enhancement in nonlinear spaces. The fuzzy methodology is further adopted in previous work. In this study, a general NFLE transformation, called fuzzy-kernel NFLE, is extended for feature extraction in which kernelization and fuzzification are simultaneously considered. In addition, more experimental analysis was conducted in this study. Three benchmark data sets were evaluated in this work. The proposed method was compared with state-of-the-art algorithms for performance evaluation.

2. Related Works

In this study, three approaches, nearest feature line embedding (NFLE), kernelization , and fuzzy *k* nearest neighbor (FKNN)[20], were considered to reduce the feature dimensions for HSI classification. Before the proposed methods, brief reviews of NFLE and kernelization methods are presented in the following: Given N *d*-dimensional training samples $X=[x_1,x_2...x_N]\in R^{d\times N}$ consisting of N_C land-cover classes $C_1,C_2,\ldots,C_{N_C}$. The new samples in a low-dimensional space were obtained by the linear projection $y_i=w^Tx_i$, where w is a found linear projection matrix for DR. NFLE is a linear transformation for DR. The sample scatters are represented in a Laplacian matrix form by using the point-to-line strategy which originated from the nearest linear combination (NLC) approach. The objective function is defined and minimized as follows:

$$\begin{aligned}O&=\sum_i\left(\sum_{i\neq m\neq n}\left\| y_i-L_{m,n}(y_i)\right\|^2 l_{m,n}(y_i)\right)\\&=\sum_i\left\|y_i-\sum_j M_{i,j}y_i\right\|^2\\&=tr\left(Y(I-M)^T(I-M)Y\right)\\&=tr\left(w^TX(D-W)w^TX\right)\\&=tr\left(w^TXLX^Tw\right).\end{aligned}\tag{1}$$

Here, point $L_{m,n}(y_i)$ is a projection point on line $L_{m,n}$ for point y_i, and weight $l_{m,n}(\mathbf{y}_i)$ (being 1 or 0) represents the connectivity relationship from point y_i to a feature line $L_{m,n}$ that passes through two points y_m and y_n. The projection point $L_{m,n}(y_i)$ is represented as a linear combination of points y_m and y_n : $L_{m,n}(y_i)=y_m+t_{m,n}(y_n-y_m)$ in which $i\neq m\neq n$, and $t_{m,n}=(y_i-y_m)^T(y_m-y_n)/(y_m-y_n)^T(y_m-y_n)$. Using simple algebra operations, the discriminant vector from point y_i to the projection point $L_{m,n}(y_i)$ can be represented as $y_i-\sum_j M_{i,j}y_j$, in which two values in the *i*th row in matrix M are set as $M_{i,m}=t_{n,m}$, $M_{i,n}=t_{m,n}$, and $t_{n,m}+t_{m,n}=1$, when weight $l_{m,n}(\mathbf{y}_i)=1$. The other values in the *i*th row are set as zero, if

$j \neq m \neq n$. The mean squared distance in Eq. (1) for all training points to their NFLs is next obtained as $tr(w^T XLX^T w)$, in which $L = D - W$, and matrix D is a matrix of the column sums of the similarity matrix W. From the consequences of Yan *et al.* [22], matrix W is defined as $W_{i,j} = (M + M^T - M^T M)_{i,j}$ when $i \neq j$, and zero otherwise; $\sum_j M_{i,j} = 1$. Matrix L in Eq. (1) is represented as a Laplacian matrix. For more details, refer to [18, 19]. Consider the class labels in supervised classification, two parameters K_1 and K_2 are manually determined in calculating the within-class scatter S_w and the between-class scatter S_b, respectively:

$$\mathbf{S}_w = \sum_{k=1}^{N_C} \left(\sum_{x_i \in C_k} \sum_{L_{m,n} \in F_{K_1}(x_i, C_k)} (x_i - L_{m,n}(x_i))(x_i - L_{m,n}(x_i))^T \right) \tag{2}$$

$$\mathbf{S}_b = \sum_{k=1}^{N_C} \left(\sum_{x_i \in C_k} \sum_{l=1, l \neq k}^{N_C} \sum_{L_{m,n} \in F_{K_2}(x_i, C_l)} (x_i - L_{m,n}(x_i))(x_i - L_{m,n}(x_i)) \right. \tag{3}$$

.

$F_{K_1}(x_i, C_k)$ indicates the set of K_1 NFLs within the same class, C_k, of point x_i, i.e. $l_{m,n}(y_i) = 1$, and $F_{K_2}(x_i, C_l)$ is a set of K_2 NFLs belonging to the different classes of point x_i. The Fisher criterion $tr(\mathrm{S}_b / \mathrm{S}_w)$ is then maximized to find the projection matrix w, which is composed of the eigenvectors with the corresponding largest eigenvalues. A new sample in the low-dimensional space can be obtained by the linear projection $y = w^T x$, and the NN (one-NN) matching rule is applied for template matching. In kernel LDA, consider the nonlinear mapping function from a space X to a Hilbert space H, $\phi : x \in X \rightarrow \phi(x) \in H$, the within-class and between-class scatter in space H are calculated as

$$\mathbf{S}_w^{\phi} = \sum_{k=1}^{N_C} \left(\sum_{x_i \in C_k} (\phi(x_i) - \bar{\phi}_k)(\phi(x_i) - \bar{\phi}_k)^T \right), \tag{4}$$

and

$$\mathbf{S}_b^{\phi} = \sum_{k=1}^{N_C} (\bar{\phi}_k - \bar{\phi})(\bar{\phi}_k - \bar{\phi})^T . \tag{5}$$

Here, $\bar{\phi}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \phi(x_i)$ and $\bar{\phi} = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i)$ represent the class mean and the population mean in space H, respectively. To generalize LDA to the nonlinear case, the dot product trick is exclusively used. The expression of dot product on the Hilbert space H is given by the following kernel function: $k(x_i, x_j) = k_{i,j} = \phi^T(x_i)\phi(x_j)$. Let the symmetric matrix K of N by N be a matrix composed of dot product in feature space H, i.e. $K(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle = (k_{i,j})$ and, $i, j = 1,2,...,N$. The kernel operator K makes the possibility: the construction of the linear separating function in space H is equivalent to that of the nonlinear separating function in space X. Kernel LDA also maximizes the between-class scatter and minimizes the within-class scatter, i.e. $\max(S_b^{\phi} / S_w^{\phi})$. This maximization is equivalent to the following eigenvector resolution: $\lambda S_w^{\phi} w = S_b^{\phi} w$. There is a set of coefficients α for $w = \sum_{i=1}^{N} \alpha_i \phi(x_i)$ such that the largest eigenvalue gives the maximum of the scatter quotient $\lambda = w^T S_b^{\phi} w / w^T S_w^{\phi} w$.

3. Fuzzy Kernel Nearest Feature Line Embedding

According to the aforementioned surveys, a training DR scheme effectively extracts the discriminant features from the non-Euclidean and non-linear space. To this end, fuzzy kernel nearest feature line embedding (FKNFLE) is proposed for HSI classification. The idea of FKNFLE is to incorporate the fuzziness and kernelization into the manifold learning method. The kernel function not only generates a non-linear feature space for well discriminant analysis, but also increases the robustness to noise during the training phase. Manifold learning methods preserve the local structure of samples in the Hilbert space. On the other hand, the fuzzy *k*-nearest neighbor (FKNN) method extracts the non-Euclidean structures of training samples for enhancing discriminative capability.

NFLE has successfully been applied in HSI classification. Noise variations and high degree non-linear data distributions limit the performance of manifold learning. A kernel trick is used to alleviate this problem. The details of FKNFLE are introduced in the following: Let $\phi : x \in X \rightarrow \phi(x) \in H$ be a nonlinear mapping from a low dimensional space to a high-dimensional Hilbert space H. The mean squared distance for all training points to their NFLs in the Hilbert space is written as follows:

$$\begin{aligned}
&\sum_i \left\| \phi(y_i) - L_{m,n}(\phi(y_i)) \right\|^{(2)} \\
&= \sum_i \left\| \phi(y_i) - \sum_j M_{i,j}\phi(y_j) \right\|^2 \\
&= tr\left(\phi^T(Y)(I-M)^T(I-M)\phi(Y)\right) \\
&= tr\left(\phi^T(Y)(D-W)\phi(Y)\right) \\
&= tr\left(w^T\phi(X)L\phi^T(X)w\right).
\end{aligned} \tag{6}$$

Then, the object function in Eq. (6) is minimized and expressed as a Laplacian matrix. The eigenvector problem of kernel NFLE in the Hilbert space is expressed as:

$$\left[\phi(X)L\phi^T(X)\right]w = \lambda\left[\phi(X)D\phi^T(X)\right]w. \tag{7}$$

To extend NFLE to its kernel version, the implicit feature vector, $\phi(x)$, does not need to be obtained explicitly. The dot product expression of two samples is exclusively applied in the Hilbert space with a kernel function as follows: $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. The eigenvectors of Eq. (7) are represented by the linear combinations of $\phi(x_1)$, $\phi(x_2), \cdots$, $\phi(x_N)$. The coefficient α_i is $w = \sum_{i=1}^{N} \alpha_i \phi(x_i) = \phi(X)\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T \in R^N$. Then, the eigenvector problem is as follows:

$$KLK\alpha = \lambda KDK\alpha. \tag{8}$$

Let the coefficient vectors, $\alpha^1, \alpha^2, \ldots, \alpha^N$, be the solutions of Eq. (8) in a column format. Given a testing point, z, the projections onto the eigenvectors, w^k, are obtained as follows:

$$\left(w^k \cdot \phi(z)\right) = \sum_{i=1}^{N} \alpha_i^k \langle \phi(z), \phi(x_i) \rangle = \sum_{i=1}^{N} \alpha_i^k K(z, x_i), \tag{9}$$

where α_i^k is the i^{th} element of the coefficient vector, α^k. The kernel function RBF (radial basis function) is used in this study. Thus, the within-class and between-class scatters in a kernel space are defined as follows:

$$\mathbf{S}_w^{\phi} = \sum_{k=1}^{N_C}\left(\sum_{\phi(x_i)\in C_k} \ \sum_{L_{m,n}\in F_{K_1}(\phi(x_i),C_k)} \left(\phi(x_i) - L_{m,n}(\phi(x_i))\right)\left(\phi(x_i) - L_{m,n}(\phi(x_i))\right)^T \right) \tag{10}$$

$$\mathbf{S}_b^{\phi} = \sum_{k=1}^{N_C}\left(\sum_{\phi(x_i)\in C_k} \ \sum_{l=1,l\neq k}^{N_C} \ \sum_{L_{m,n}\in F_{K_2}(\phi(x_i),C_l)} \left(\phi(x_i) - L_{m,n}(\phi(x_i))\right)\left(\phi(x_i) - L_{m,n}(\phi(x_i))\right)^T \right). \tag{11}$$

The kernelized manifold learning could preserve the non-linear local structure in a Hilbert space. The distances in the NFLE approach are calculated by the Euclidean distance-based measurement. On the other hand, the non-Euclidean structure of training samples can be further extracted by fuzzification. The FKNN algorithm[20] enhances the discriminant power among samples by assigning the higher membership grades to the samples whose neighbors are with the same class. By doing so, the non-Euclidean structures are extracted, and the discriminative power of samples can be enhanced. The idea of FKNFLE using the fuzzification trick is described in the following.

Consider N samples in the reduced space $Y = [y_1, y_2 \ldots, y_N]$ and their corresponding fuzzy membership grades, $\pi(y_i)$, for each sample, y_i. The objective function is re-defined as follows:

$$\begin{aligned}
O &= \sum_i \pi(y_i)\left(\sum_{i\neq m\neq n} \left\| y_i - L_{m,n}(y_i) \right\|^2 l_{m,n}(y_i) \right) \\
&= \sum_i \pi(y_i) \left\| y_i - \sum_j M_{i,j} y_j \right\|^2 \\
&= tr\left(Y^T (FEI - FEM)^T (FEI - FEM) Y\right) \\
&= tr\left(Y^T (FED - FEW) Y\right) \\
&= tr\left(Y^T \left(D_{fuzzy} - W_{fuzzy}\right) Y\right) \\
&= tr\left(w^T X L_{fuzzy} X^T w\right)
\end{aligned} \tag{10}$$

Here, each sample is assigned a fuzzy grade, $\pi(y_i)$. Element $M_{i,j}$ denotes the connectivity relationship between point y_i and line $L_{m,n}$ which is the same as that in Eq. (1). Two non-zero terms, $M_{i,n} = t_{m,n}$ and $M_{i,m} = t_{n,m}$, are set, and $\sum_j M_{i,j} = 1$. Using simple algebra operations, the objective function with fuzzification is represented in a Laplacian matrix form in which the fuzzy terms, $\pi(y_i)$, constitute the column vector, F ,with size $N \times 1$, and E is a row vector of all those with size $1 \times N$.

Similarly, given N samples $\phi(\mathrm{X}) = \{\phi(x_1), \phi(x_2), \ldots, \phi(x_N)\}$ in a Hilbert space, the membership grade of a specified sample, $\phi(x_i)$, and its K_3 neighbors is designed in the following equation for computing the within-class scatter:

$$\pi(x_i) = \begin{cases} 0.51 + \left(0.49 * (q_i / K_3)\right), & \text{if } q_i \geq \theta_{within}; \\ \\ 0.49 * (q_i / K_3) & \text{otherwise.} \end{cases} \tag{11}$$

Here, value q_i is the number of samples whose labels are the same label of $\phi(x_i)$ among K_3 nearest neighbors, and θ_{within} is a manual threshold. If $q_i = K_3$, then $\pi(x_i)$ returns to 1, i.e. all neighbors are in the same class. Adding the fuzzy term $\pi(x_i)$, the within-class scatter matrix becomes:

$$\mathbf{S}_w^{\phi F} = \sum_{k=1}^{N_C} \left(\sum_{\phi(x_i)\in C_k} \pi(x_i) * \sum_{L_{m,n}\in F_{K_1}(\phi(x_i),C_k)} \big(\phi(x_i) - L_{m,n}(\phi(x_i))\big)\big(\phi(x_i) - L_{m,n}(\phi(x_i))\big)^T \right) \tag{14}$$

Similarly, a fuzzy term $\lambda(x_i)$ is also adopted to evaluate the membership grade of $\phi(x_i)$ and its neighbors during the computation of between-class scatter as follows:

$$\lambda(x_i) = \begin{cases} 0.51 + \big(0.49 * (p_i / K_4)\big) & \text{if } p_i \geq \theta_{between}; \\ \\ 0.49 * (p_i / K_4) & \text{otherwise.} \end{cases} \tag{12}$$

Here, value p_i is the number of samples with labels different from $\phi(x_i)$ among K_4 nearest neighbors, and $\theta_{between}$ is a given threshold. If $p_i = K_4$, term $\lambda(x_i)$ is returned to 1. That means all neighbors have labels different from $\phi(x_i)$. The fuzzy term $\lambda(x_i)$ is added into the between-class scatter matrix to generate a new one as:

$$\mathbf{S}_b^{\phi F} = \sum_{k=1}^{N_C} \left(\sum_{\phi(x_i)\in C_k} \lambda(x_i) * \sum_{l=1,l\neq k}^{N_C} \sum_{L_{m,n}\in F_{K_2}(\phi(x_i),C_l)} \big(\phi(x_i) - L_{m,n}(\phi(x_i))\big)\big(\phi(x_i) - L_{m,n}(\phi(x_i))\big)^T \right) \tag{16}$$

Hence, kernelization and fuzzification are simultaneously integrated into the NFLE transformation for feature extraction. In this paper, a general format for the NFLE learning method using kernelization and fuzzification is proposed to be used for DR. The advantages of the proposed method are threefold: the kernelization strategy generates a non-linear feature space for the discriminant analysis and increases the robustness to noise for manifold learning; the kernelized manifold learning preserves the local manifold structure in a Hilbert space as well as the locality of the manifold structure in the reduced low dimensional space; non-Euclidean structures are extracted for improving discriminative abilities using the FKNN strategy.

4. Experimental Results

In this section, the experimental results are discussed to demonstrate the effectiveness of the proposed method for HSI classification. Three HSI benchmarks are given for evaluation. The first data set, Indian Pines Site (IPS) image, was generated from AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) which was captured by the Jet Propulsion Laboratory and NASA/Ames in 1992. The IPS image was captured from 6 miles in the western area of the *Northwest Tippecanoe County* (NTC). A false color IR image of dataset IPS is shown in Fig. 2(a). Although dataset IPS contained 220 bands and 16 land-cover classes, only ten classes were used in the experiments: *Corn-no till*(1428), *Corn-min till*(830), *Grass/Pasture*(478), *Grass/Trees*(730), *Hay-windrowed*(483), *Soybeans-no till*(972), *Soybeans-min till*(2455), *Soybeans-clean till*(593), *Woods*(1265), and *Bldg-Grass-Tree-Drives*(386). The numbers in parentheses are the collected pixel numbers in dataset IPS. The ground truths of 9,620 pixels were manually labeled for training and testing. Nine hundred training samples of ten classes were randomly chosen from 9,620 pixels, and the remaining samples were used for testing. The other two HSI data sets adopted in the experiments were obtained from the Reflective Optics System Imaging Spectrometer (ROSIS) instrument covering the City of Pavia, Italy. Two scenes, the university area and the Pavia city centre containing 103 and 102 data bands both with a spectral coverage from 0.43 to 0.86 um and a spatial resolution of 1.3m. The image sizes of these two areas were 610x340 and 1096x715 pixels, respectively. Figs. 2(b) and 2(c) show the false color IR image of these two data sets. Nine land-cover classes were available in each data set, and the samples in each data set were separated into two subsets, i.e. one training and one testing set. Given the Pavia University data set, ninety training samples per class were randomly collected for training, and the 8,046 remaining samples were tested for performance evaluation. Similarly, the numbers of training and testing samples used for the Pavia Centre data set were 810 and 9,529, respectively. The proposed methods, NFLE[18,19], KNFLE, FNFLE[24], and FKNFLE, were compared with two state-of-the-art algorithms, i.e. nearest regularized subspace (NRS) [23] and NRS-LFDA [23]. The parameter configurations for both algorithms NRS[1] and NRS-LFDA can be referred to in [23]. The gallery samples were randomly chosen for training the transformation matrix, and the query samples were matched with the gallery samples using the NN matching rule. Each algorithm was run 30 times to obtain the average rates. To obtain the appropriate reduced dimensions of FKNFLE, the available

[1] The source codes are available from the web site https://github.com/eric-tramel/NRSClassifier

training samples were chosen to evaluate the overall accuracy versus the reduced dimensions in the benchmark datasets. The proposed method was compared with various classification methods on computational time. All methods were implemented by MATALB codes on a personal computer with an i7 2.93-GHz CPU and 12.0 GB RAM. The comparisons of various algorithms on computational time were tabulated in Table I for the IPS, Pavia University, and Pavia City Centre datasets. Considering the training time, the proposed FKNFLE algorithm was generally faster than NRS and NRS-LFDA; 2 times and 15 times, respectively. Due to the fuzzification process, algorithms FKNFLE and FNFLE were slower than KNFLE and NFLE; 13 times and 15 times, respectively.

From Tables II to IV, the producer's accuracy, overall accuracy, kappa coefficients, and user's accuracy defined by the error matrices (or confusion matrices) [25] were calculated for performance evaluation. They are briefly defined in the following.

The user's accuracy and the producer's accuracy are two widely used measures for class accuracy. The user's accuracy is defined as the ratio of the number of correctly classified pixels in each class by the total pixel number classified in the same class. The user's accuracy is a measure of commission error, whereas the producer's accuracy measures the errors of omission and indicates the probability that certain samples of a given class on the ground are actually classified as such. The kappa coefficient, also called the kappa statistic, is defined to be a measure of the difference between the actual agreement and the changed agreement.

Table I: The training and testing time of various algorithms for the benchmark datasets (seconds).

Datasets	IPS		Pavia University		Pavia City Centre	
Algorithms	Training	Testing	Training	Testing	Training	Testing
	900	8720	810	8046	810	9529
NFLE-NN	10	18	9	16	9	20
KNFLE-NN	12	18	11	16	11	20
FNFLE-NN	155	18	140	16	140	20
FKNFLE-NN	156	18	141	16	141	20
NRS	326	326	294	300	294	351
LFDA-NRS	2331	327	2098	301	2098	352

Table II: The classification error matrix for data set IPS (in percentage).

Classes	Reference Data										User's Accuracy
	1	2	3	4	5	6	7	8	9	10	
1	79.20	3.43	0.28	0.35	0	5.46	9.73	1.54	0	0	79.20
2	5.90	81.81	0	0.12	0	1.33	6.39	4.34	0	0.12	81.81
3	0	0	97.49	1.46	0.21	0.42	0	0.21	0.42	0.84	97.49
4	0	0	0.27	96.30	0	0	0	0	0	3.42	96.30
5	0	0	0.42	0	99.58	0	0	0	0	0	99.58
6	5.14	0.21	0.10	0.41	0	88.89	4.42	0.72	0	0.10	88.89
7	10.59	5.58	0.29	0.33	0.04	9.78	69.98	3.30	0	0.12	69.98
8	1.35	4.05	1.52	0.34	0	1.69	1.85	88.53	0	0.67	88.53
9	0	0	3.32	0.16	0	0	0	0	90.83	5.69	90.83
10	0	0	3.89	5.70	0	0	0	0.26	10.88	79.27	79.27
Producer's Accuracy	77.51	86.04	90.62	91.57	99.75	82.63	75.76	89.51	88.94	87.85	

Kappa Coefficient: **0.821** Overall Accuracy: **83.34%**

Table III: The classification error matrix for data set Pavia University (in percentage).

Classes	Reference Data									User's Accuracy
	1	2	3	4	5	6	7	8	9	
1	90.18	3.15	0	0	0	3.24	1.35	1.26	0.81	90.18
2	2.31	92.50	0	2.31	0	1.85	0	1.01	0	92.50
3	0	0	90.07	2.38	1.58	0.99	2.97	0.99	0.99	90.07
4	0	1.23	2.84	90.24	1.42	1.42	1.51	1.32	0	90.24
5	0.63	1.13	0.75	1.26	91.91	0.63	1.64	0.88	1.13	91.91
6	1.10	1.19	1.38	1.56	1.19	92.54	0.55	0.46	0	92.54
7	0	1.12	0.51	0.61	2.24	0	93.25	1.22	1.02	93.25
8	0.47	1.42	0.95	1.42	2.38	1.90	0	90.76	0.66	90.76
9	1.14	0	2.15	2.01	0	2.29	0	2.15	90.22	90.22
Producer's Accuracy	94.10	90.92	91.30	88.65	91.25	88.25	92.08	90.71	95.14	

Kappa Coefficient: **0.910** Overall Accuracy: **91.31%**

Table IV: The classification error matrix for data set Pavia City Centre (in percentage).

Classes	Reference Data									User's Accuracy
	1	2	3	4	5	6	7	8	9	
1	98.61	0.17	0.51	0.34	0.34	0	0	0	0	98.61
2	1.04	97.47	0.43	0	0	0.34	0.17	0.52	0	97.47
3	0.59	0.82	96.23	0.69	0.99	0	0	0	0.69	96.23
4	0	0.56	0.66	96.68	0.37	0.47	0.66	0.56	0	96.68
5	0	0	0.43	0.34	97.73	0.26	0.34	0.34	0.52	97.73
6	0.35	0.26	0.61	0	0	98.15	0	0.26	0.35	98.15
7	0.35	0.26	0	0.35	0	0.44	98.23	0.35	0	98.23
8	0	0	0.37	0.30	0.37	0.52	0.45	97.43	0.52	97.43
9	0.39	0.59	0.79	0.29	0.29	0	0	0	97.60	97.60
Producer's Accuracy	97.32	97.34	96.20	97.67	97.64	97.97	98.38	97.96	97.91	

Kappa Coefficient: **0.971** Overall Accuracy: **97.59%**

5. SECTIONS

In this paper, a general NFLE transformation, FKNFLE, for HSI classification is proposed. Kernelization and fuzzification were both considered in NFLE in extracting non-linear and non-Euclidean structures. In addition, the locality of the manifold structure of samples was preserved. High-dimensional HSI data were reduced to low-dimensional features by the proposed FKNFLE transformation. Two state-of-the-art algorithms, NRS and NRS-LFDA, were compared with the proposed FKNFLE. Three land-cover benchmarks, IPS, Pavia University, and Pavia City Centre, were tested for performance evaluation. The experimental results demonstrated that FKNFLE outperformed the other algorithms.

6. REFERENCES

M. Turk and A.P. Pentland, “Face recognition using eigenfaces,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1991, pp. 586-591.

P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Ana Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711-720, Jul. 1997.

H. Cevikalp, M. Neamtu, M. Wikes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. Pattern Ana Anal. Mach. Intell.*, vol. 27, no. 1, pp. 4-13, Jan. 2005.

S. Prasad and L. Mann Bruce, "Information fusion in kernel-induced spaces for robust subpixel hyperspectral ATR," *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 6, no. 3, pp. 572-576, Jul. 2009.

X. He, S. Yan, Y. Ho, P. Niyogi, and H. J. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Ana Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328-340, Mar. 2005.

S. T. Tu, J. Y. Chen, W. Yang, and H. Sun, "Laplacian eigenmaps-based polarimetric dimensionality reduction for SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 1, pp. 170-179, Jan. 2011.

Z. Wang and B. He, "Locality preserving projections algorithm for hyperspectral image dimensionality reduction," in *Proc. 19th Int. Conf. Geoinf.*, Jun. 24-26, 2011, pp. 1-4.

D. H. Kim, and L. H. Finkel, "Hyperspectral image processing using locally linear embedding," in *Proc. 1st Int, IEEE EMBS Conf. Neural Eng.*, Italy, Mar. 20-22, 2003, pp. 316-319.

W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification," *IEEE Geosci. Remote Sens. Letters*, vol. 8, no. 5, pp. 894-898, Sep. 2011.

W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.

R. B. Luo, W. Z. Liao, and Y. G. Pi, "Discriminative supervised neighborhood preserving embedding feature extraction for hyperspectral-image classification," *Telkomnika*, vol. 10, no. 5, pp. 1051–1056, 2012.

B. Boots, and G. J. Gordon, "Two-manifold problems with applications to nonlinear system Identification," *Proc. 29th International Conference on Machine Learning.*, United Kindom, Jun. 26- Jul. 1, 2012.

F. Odone, A. Barla, and A. Verri, "Building kernels from binary strings for image matching," *IEEE Trans. Image Processing.*, vol. 14, no. 2, pp.169-180, Feb. 2005.

B. Scholkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation.*, vol. 10, no. 5, pp.1299-1319, Jul. 1998.

Y. Y. Lin, T. L. Liu, and C. S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Trans. Pattern Analysis. Mach Intell.*, vol. 33, no. 6, pp.1147-1160, Jun. 2011.

J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp.4816-4829, Sep. 2013.

Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp.217-231, Jan. 2013.

Y. N. Chen, C. C. Han, C. T. Wang, and K. C. Fan, "Face recognition using nearest feature space embedding," *IEEE Trans. Pattern Analysis. Mach Intell.*, vol. 33, no. 6, pp.1073-1086, Jun. 2011.

Y. L. Chang, J. N. Liu, C. C. Han, and Y. N. Chen, "Hyperspectral image classification using nearest feature line embedding approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 278–287, Jan. 2014.

J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, no. 4, pp.580-585, July/August. 1985.

S. Z. Li, "Face recognition based on nearest linear combinations," *Proc. Computer Vision and Pattern Recognition*, pp. 839-844, 1998.

S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a framework for dimensionality reduction," *IEEE Trans. Pattern Analysis. Mach Intell.*, vol. 29, no. 1, pp.40-51, Jun. 2007.

W. Li, E. W. Tramel, S. *Prasad*, and J. E. Fowler, "Nearest regularized subspace for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 477–489, Jan. 2014.

Y. N. Chen, C. C. Han, and K. C. Fan, "Use fuzzy nearest feature line embedding for hyperspectral image classification," *Proc. 4th International Conf. Earth Observations and Societal Impacts*, Taiwan, Jun. 22-24, 2014.

T. M. Lillesand and R. W. Kiefer, *Remote Sensing and Image Interpretation*. New York: Wiley, 2000.

Over- and Under-Segmentation Evaluation based on the Segmentation Covering Measure

Jose Sigut
University of La Laguna
Department of Computer Science
Faculty of Physics
Spain (38200), La Laguna, Tenerife
sigut@isaatc.ull.es

Francisco Fumero
University of La Laguna
Department of Computer Science
Faculty of Physics
Spain (38200), La Laguna, Tenerife
franfumero@isaatc.ull.es

Omar Nuñez
University of La Laguna
Department of Computer Science
Faculty of Physics
Spain (38200), La Laguna, Tenerife
omar@isaatc.ull.es

ABSTRACT

Very few measures intended for evaluating the quality of image segmentations account separately for over- and under-segmentation. This distinction is highly desirable in practice because in many applications under-segmentation is considered as a much serious issue than over-segmentation. In this paper, a new approach to this problem is presented as a decomposition of the Segmentation Covering measure into two contributions, one due to over-segmentation and the other one to under-segmentation. Our proposal has been tested on the output of state-of-the-art segmentation algorithms using the Berkeley image database. The results obtained are comparable to those provided by similar evaluation methods allowing a clear separation between over- and under-segmentation effects.

Keywords

Image segmentation, segmentation evaluation, over-segmentation, under-segmentation.

1. INTRODUCTION

Image segmentation plays a major role in a broad range of computer vision applications. Therefore, there is a strong need for objective measures of the quality of a segmentation algorithm on an image or set of images. The most usual way to accomplish this task is by comparing the segmentation at hand with a set of manually-segmented reference images which are often referred as gold standard or ground truth. In recent years there has been a great effort to provide adequate evaluation measures and image databases which have been used as gold standards for different applications [Mar01a] [Unn07a]. However, hardly any of these measures accounts explicitly for over- and under-segmentation. This distinction is highly desirable in practice because in many applications under-segmentation is considered as a much serious problem than over-segmentation since it is usually easier to merge segments to obtain bigger ones than splitting large regions to recover the true segments.

The Segmentation Covering measure has been proven to be a good choice for evaluating segmentation performance [Arb11a]. We will show that under mutual refinement this measure can be written as the contribution of two terms, one of them dealing with over-segmentation and the other one with under-segmentation. An extension to the more general case of arbitrary overlapping regions is also provided. The proposed evaluation method has been tested on the output of three state-of-the-art segmentation algorithms and compared with other evaluation measures using the well-known Berkeley image database [Mar01a].

The rest of the paper is organized as follows. Section II is about related approaches to deal separately with over- and under-segmentation. Section III describes the Segmentation Covering measure and the proposed evaluation method which is derived from it. The experimental results are shown and discussed in section IV. Section V is devoted to the conclusions.

2. RELATED WORK

As far as we know, there are few approaches which account separately for over- and under-segmentation as compared to global evaluation measures. Cardoso and Corte-Real [Car05a] introduce the concept of partition distance $d_{sym}(G,S)$ between a reference segmentation G and the segmentation under study S as a symmetric measure and propose to use an asymmetric version $d_{asy-ov}(G,S)$ for the case of applications where over-segmentation is not an issue. An analogous asymmetric measure $d_{asy-un}(S,G)$ is proposed for the case of under-segmentation.

The information based distance VI proposed by Meila [Meiqq07a] is one of the most popular evaluation measures and is given by

$$VI(S,G) = H(S) + H(G) - 2I(S,G) \qquad (1)$$

Where H and I, respectively, represent the entropies and mutual information between S and G.

Meila shows that VI can be written as the sum of two conditional entropies

$$VI(S,G) = H(S|G) + H(G|S) \qquad (2)$$

Where the conditional entropies $H(S|G)$ and $H(G|S)$ are identified by Gong and Shi [Gon11a] as over- and under-segmentation metrics, respectively.

Other researchers have focused only on the under-segmentation error. Levinshtein et al [Lev09a] compute this error by means of

$$U_{E-TP} = \frac{1}{numG} \sum_{R_i \in G} \frac{\left[\sum_{S_j \in S: S_j \cap R_i \neq \emptyset} |S_j|\right] - |R_i|}{|R_i|} \qquad (3)$$

Where $numG$ is the number of regions in G, R_i denotes any region belonging to G and S_j denotes any region belonging to S. The main disadvantage of using (3) is that it tends to overestimate the amount of under-segmentation because of the inclusion in the calculation of large regions in S with very little overlap. In order to avoid this, Achanta et al [Ach12a] suggest a similar error measure but restricting the overlap to be at least a certain percentage of the segment size as it is expressed in

$$U_{E-Slic} = \frac{1}{N} \sum_{R_i \in G} \sum_{S_j \in S: |S_j \cap R_i| > B} |S_j| \qquad (4)$$

Where N is the image size and B is the specified percentage which is set by the authors to 5%.

Protzel and Neubert [Pro12a] propose an alternative under-segmentation measure which overcomes the need for additional parameters. They define the under-segmentation error as

$$U_e = \frac{1}{N} \sum_{R_i \in G} \sum_{S_j \in S: S_j \cap R_i \neq \emptyset} min\left(S_{jin}, S_{jout}\right) \qquad (5)$$

Where S_{jin} is the portion of S_j inside R_i and S_{jout} is the portion of S_j outside R_i

3. SEGMENTATION COVERING AND PROPOSED MEASURES

The classic overlap measure between two regions R and R' is given by:

$$O(R,R') = \frac{|R \cap R'|}{|R \cup R'|} \qquad (6)$$

The Segmentation Covering measure introduced by Arbelaez et al [Arb09a] can be seen as a generalization of (6) to multiple regions so that the covering of a reference segmentation G by a segmentation S is defined as

$$SC(G,S) = \frac{1}{N} \sum_{R_i \in G} |R_i| max O\left(R_i, S_j\right)_{S_j \in S} \qquad (7)$$

The definition in (7) can be extended to a family of ground truth segmentations $\{G_i\}$ by first covering each G_i separately with S, and then averaging over them. It can also be analogously defined the covering of S by $\{G_i\}$ but in what follows we will assume that the segmentation covering is calculated as in (7).

Let us consider the ideal case of mutual refinement between the ground truth segmentation and the segmented image. G is said to be a mutual refinement of S if the intersection of every region R_i of G with every region S_j of S is either empty or equal to any of them. From the definition, it is easy to see that if G is a mutual refinement of S, then S is a mutual refinement of G. Figure 1 shows a trivial example of mutual refinement between two images. Under this assumption, it can be shown that each term in the summation in (7) will contribute to the final covering with either over-segmentation or under-segmentation.

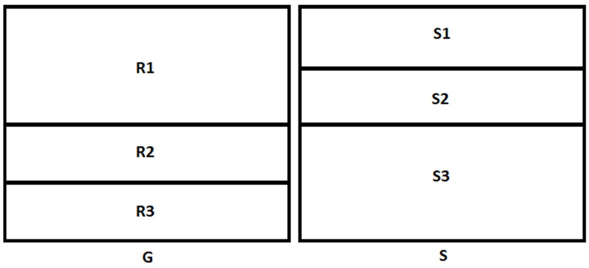


Figure 1. Example of mutual refinement between G and S

In the case of over-segmentation, according to Figure 2, it is clear that

$$O\left(R_i, S_j\right)_{S_j \in S} = \frac{|S_j|}{|R_i|} \qquad (8)$$

Therefore, the whole contribution can be simply written as

$$|R_i| max O\left(R_i, S_j\right)_{S_j \in S} = max|S_j| \qquad (9)$$

In the case of under-segmentation there must be at least two regions of G, R_1 and R_2, contained in a region of S, as shown in Figure 2. It is clear that in this situation the overlap is already maximum so that

$$|R_i|maxO\left(R_i,S_j\right)_{S_j\in S}=\frac{|R_i|^2}{|S_j|}, i=1,2 \qquad (10)$$

By adding the two terms, we obtain:

$$\sum_{i=1,2}|R_i|maxO\left(R_i,S_j\right)_{S_j\in S}=\frac{|R_1|^2+|R_2|^2}{|S_j|} \qquad (11)$$

The expression in (11) can be easily generalized to an arbitrary number of regions. From the exposed above, (7) can be written as

$$SC(G,S)=SC_{ov}+SC_{un} \qquad (12)$$

Where SC_{ov} and SC_{un} are defined respectively as the over- and under-segmentation contributions to the Segmentation Covering and given by

$$SC_{ov}(G,S)=\frac{1}{N}\sum_{R_i\in G} max|S_j|_{S_j\subseteq R_i} \qquad (13)$$

$$SC_{un}(G,S)=\frac{1}{N}\sum_{S_j\in S}\frac{\sum_{R_i\subset S_j}|R_i|^2}{|S_j|} \qquad (14)$$

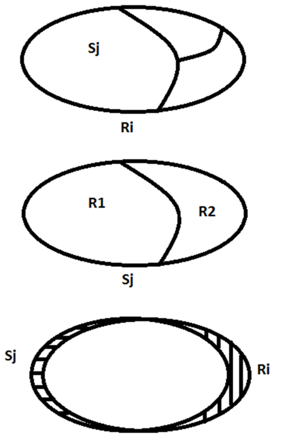


Figure 2. The top image shows over-segmentation under the mutual refinement assumption, the image in the middle shows under-segmentation, and the bottom image shows a more realistic situation of arbitrary overlap

According to (13), in the case of perfect overlap, i.e. $G=S$, $SC_{ov}(G,S)=SC(G,S)$. On the other hand, if the assumption of mutual refinement is not met, as it is usually the case, the expressions in (13) and (14) are not adequate to compute over- and under-segmentation. Figure 2 shows an example of a more realistic scenario of overlap between two segments. Each region is mostly contained in the other one but not completely so it is not clear how over- and under-segmentation should be measured in such a situation.

Our proposal consists of setting a threshold parameter to determine which contribution to the covering in (7) should be considered as either over- or under-segmentation. More concretely, given a region belonging to the ground truth R_i, a segment S_j will be seen to contribute to over-segmentation in that region as long as

$$|R_i\cup S_j|\leq|R_i|+\gamma|R_i| \qquad (15)$$

So that the amount of pixels outside R_i to be considered as over- or under-segmentation is controlled by the γ parameter. If every segment S_j which overlaps with a region R_i satisfies (15), the contribution to over-segmentation will be equal to the covering itself for that region. The under_segmentation contribution can be simply defined as the difference between the covering and over_segmentation values. Thus, we can write

$$SC_{ov}(G,S)=\frac{1}{N}\sum_{R_i\in G}|R_i|maxO\left(R_i,S_j\right)_{S_j:|R_i\cup S_j|\leq|R_i|+\gamma|R_i|} \qquad (16)$$

$$SC_{un}(G,S)=SC(G,S)-SC_{ov}(G,S) \qquad (17)$$

By setting γ=0, (16) and (17) become equivalent to (13) and (14) under the assumption of mutual refinement. SC_{ov} and SC_{un} can be either used as absolute measures as they appear in (16) and (17) or as relative measures given by

$$SC_{ovrel}=\frac{SC_{ov}}{SC},\ SC_{unrel}=\frac{SC_{un}}{SC} \qquad (18)$$

As it will be shown in the next section, the relative measures provide a convenient means of evaluating over- and under-segmentation.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Some experiments have been carried out to show the performance of the proposed measures with respect to other measures mentioned in section II. First of all, we will focus on how to set the γ parameter in (15). In general, γ can be set to any positive or zero value depending on the application at hand but in this section we propose a more neutral procedure independent of any particular application or segmentation algorithm.

The proposed procedure is based solely on the reference segmentations provided by the Berkeley image database. Each of the 500 images has an associated ground truth consisting of between 4 and 9 hand-labeled images. The average segmentation covering among these reference images has been computed as well as the average SC_{ov} over them for different values of γ. Figure 3 shows the results of the computation sorted by the average covering value in ascending order, i.e. the agreement among humans for the different images according to this evaluation measure. For the sake of clarity, only the part of the curve with a covering value above 0.9 is shown, corresponding to those ground truth for which there is a strong agreement among subjects. Under these circumstances, very little under-segmentation can be expected and the values of SC_{ov} should be very close to the covering values. According to Figure 3, in order to comply with this requirement, the value chosen for γ should be above 0.25, otherwise it turns out to be too sensitive to small deviations from perfect overlap. For this reason, in all our experiments the value of γ was set to 0.25.

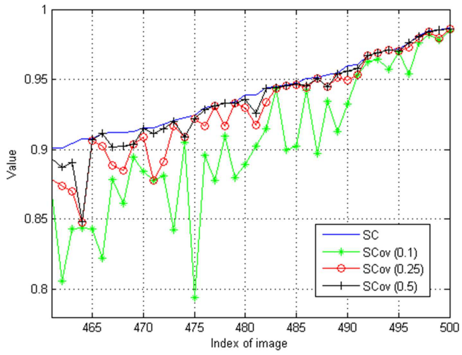


Figure 3. SC and SCov for different values of γ

For the purpose of performance comparison, the asymmetric distances $d_{asy-ov}(G,S)$ and $d_{asy-un}(S,G)$, the conditional entropies $H(S|G)$ and $H(G|S)$, and the under-segmentation error U_e have been selected. The evaluation measures have been tested on the output of three state-of-the-art segmentation methods: the OWT-UCM [Arb11a], the Mean-Shift algorithm [Com02a], and the Efficient Graph segmentation method [Fel04a]. The OWT-UCM has only one threshold parameter to be set which was varied in the range 0<level<1. The Mean-Shift algorithm has three free parameters: color range hr, spatial range hs, and minimum region size minsizeMS. It is well known that the most influential one is hr and for this reason we have set the two others to constant values hs=25, minsizeMS=10, and varied hr in the range 1<hr<30. The Efficient Graph segmentation method has also three parameters and as it happens with the Mean-Shift algorithm, one of them is more influential than the others. Following [Pen13a], we have set the alpha and minimum region size parameters to constant values: alpha=0.5, minsizeEG=10, and let the K parameter vary in the range 100<K<3000. It is very important to remark that the ranges for the parameters of the different methods have been chosen to provide segmentations at varying granularities, from strong over-segmentation with a lot of small regions to strong under-segmentation with very few segments or even just one.

Figures 4, 5, 6, 7, 8 and 9 show the values of the selected over- and under-segmentation evaluation measures averaged over the 500 images of the Berkeley database for the three segmentation algorithms at the specified parameters. The curves corresponding to the conditional entropies have been scaled to the range [0, 1] using the bounds provided in [Gon11a], $log2(N) - H\{G\}$ for the over-segmentation entropy and $H\{G\}$ for the under-segmentation entropy ($H\{G\}$ being the entropy of G and N defined as in (4)), so that they can be more easily compared to the other measures.

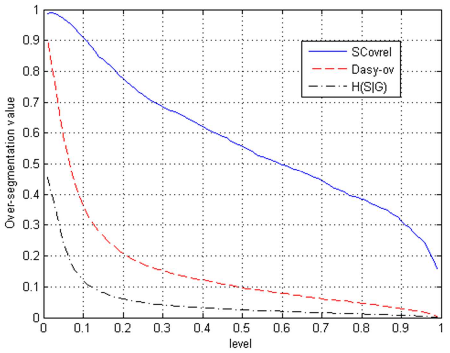


Figure 4. Average over-segmentation values for OWT-UCM

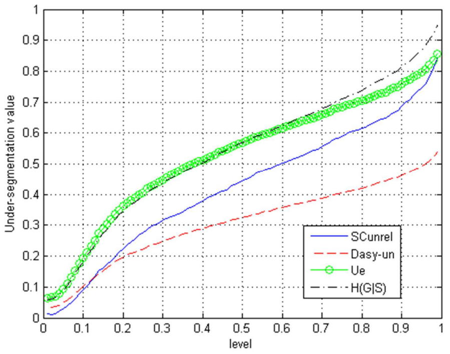


Figure 5. Average under-segmentation values for OWT-UCM

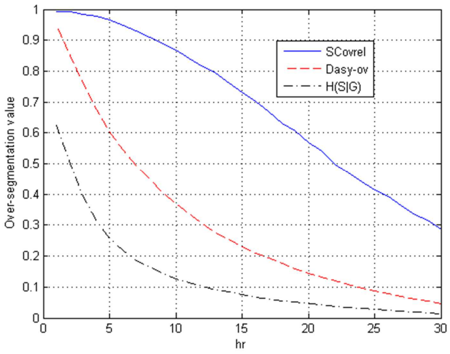


Figure 6. Average over-segmentation values for Mean Shift

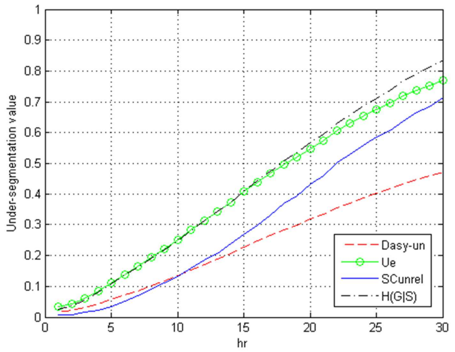


Figure 7. Average under-segmentation values for Mean Shift

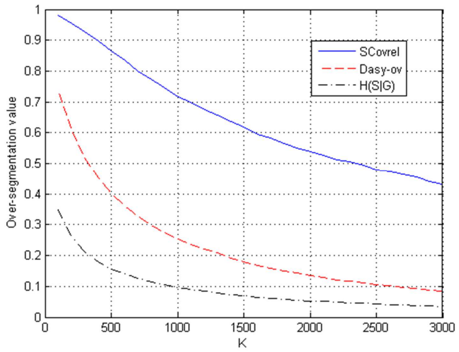


Figure 8. Average over-segmentation values for Efficient Graph

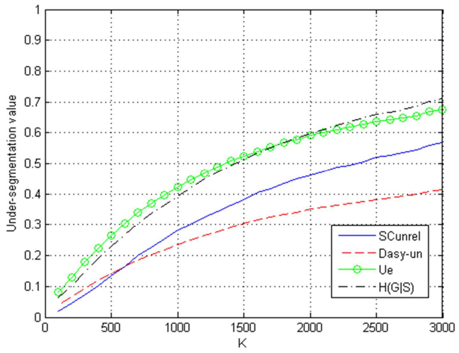


Figure 9. Average under-segmentation values for Efficient Graph

It is difficult to establish a fair comparison among the results obtained for the different measures since they are strongly dependent on how these measures should be interpreted according to their definition. In any case, a high dynamic range to clearly distinguish over-segmentation from under-segmentation seems a reasonable requirement.

In what respects to average over-segmentation, $H(S|G)$ and D_{asy-ov} decrease faster than SC_{ovrel}. As it was already pointed out, SC_{ovrel} measures the relative amount of over-segmentation and takes high values for any over-segmented image including the perfect overlap (good segmentation) as an extreme case. For this reason, it should be considered in conjunction with the covering value itself SC so that it can be correctly interpreted. The dynamic range of $H(S|G)$ is lower than the other two measures, in particular for the UCM-OWT algorithm where the rate of change in the granularity of the segmentations is higher than in the other two algorithms.

Concerning average under-segmentation, the behavior of $H(G|S)$ is very similar to U_e showing a high dynamic range. SC_{unrel} provides also a high dynamic range. Particularly remarkable are the values obtained for the different measures at the upper bound of the interval in the OWT-UCM algorithm where segmentations with only one region are common. Despite this extreme under-segmentation, the average value for D_{asy-un} is only around 0.5 (half the scale).

Table 1 shows the values of the different evaluation measures for certain images at different levels of granularity (OWT-UCM algorithm computed at levels 0.05, 0.5 and 0.9). The images are shown in Figure 10 in the appendix together with the corresponding ground truth. The results are, in general, in accordance with the average curves. The proposed measures clearly separate the over- and under-segmentation effects as it can be seen, for example, in image 6. The image is over-segmented for level=0.05 and consequently SC_{ovrel}=1 as opposed to what happens for level=0.9 where there is only one segment so that SC_{unrel}=1. For level=0.5, the tiger and part of the prey are still correctly segmented but some large parts of the image are not, leading to moderate under-segmentation and this is reflected in the values of SC_{ovrel}=0.29 and SC_{unrel}=0.71.

5. CONCLUSIONS AND FUTURE WORK

Two new evaluation measures have been proposed for dealing separately with over- and under-segmentation. They have been obtained as a decomposition of the Segmentation Covering measure in two contributions. The results of the experiments carried out have been satisfactory showing a good agreement between the values taken by the proposed measures and what should be clearly considered as over- or under-segmentation. It seems that this approach could be also used as a global segmentation evaluation methodology and this is the aim of our future work.

	Evaluation measures							
	SC	SC_{ovrel}	SC_{unrel}	$D_{asy\text{-}ov}$	$D_{asy\text{-}un}$	$H(S\|G)$	$H(G\|S)$	U_e
1	0.53	1	0	0.47	0	2.46	0.03	0.01
	0.99	1	0	0	0	0.03	0.03	0.01
	0.99	1	0	0	0	0.03	0.03	0.01
2	0.42	1	0	0.57	0.05	3.17	0.23	0.10
	0.40	0.63	0.37	0.15	0.46	0.59	1.87	0.70
	0.16	0.34	0.66	0.03	0.77	0.11	3.59	0.94
3	0.30	0.80	0.20	0.64	0.09	3.50	0.33	0.18
	0.43	0.54	0.46	0.03	0.48	0.13	1.81	0.76
	0.18	0	1	0	0.75	0	2.75	1
4	0.62	0.88	0.12	0.30	0.10	1.81	0.42	0.21
	0.23	0.23	0.77	0.05	0.63	0.27	3.06	0.90
	0.08	0.11	0.89	0.02	0.83	0.12	4.08	0.98
5	0.42	1	0	0.58	0.04	2.99	0.17	0.08
	0.70	0.32	0.68	0.08	0.20	0.25	0.79	0.40
	0.35	0	1	0	0.48	0	1.78	0.97
6	0.22	1	0	0.78	0.02	5.18	0.09	0.04
	0.51	0.29	0.71	0.05	0.38	0.16	1.32	0.76
	0.33	0	1	0	0.53	0	1.93	1
7	0.69	1	0	0.30	0.03	2.07	0.15	0.06
	0.84	1	0	0.12	0.04	0.42	0.27	0.08
	0.59	0	1	0	0.28	0	0.91	0.56

Table 1. Evaluation measures calculated for the segmented images in Figure 10. There are three values for each measure corresponding to levels 0.05, 0.5 and 0.9, from top to bottom in that order. The image index is shown on the left

6. ACKNOWLEDGMENTS

This research has been supported by an FPU grant from the Spanish Government (AP2012-6800), and has also been partially funded by the Project PI12/02301 from the Fondo de Investigación Sanitaria, Instituto Carlos III (co-financed with Feder funds).

7. REFERENCES

[Ach12a] Achanta, R., Shaji, A., Smith, K., et al., SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34, no. 11, pp. 2274–2282, 2012.

[Arb09a] Arbelaez, P., Maire, M., Fowlkes, C., et al., From contours to regions: An empirical evaluation. In IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2009, pp. 2294–2301.

[Arb11a] Arbelaez, P., Maire, M., Fowlkes, C., et al., Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33, no. 5, pp. 898–916, 2011.

[Car05a] Cardoso, J., and Corte-Real, L., Toward a generic evaluation of image segmentation. IEEE Transactions on Image Processing, 14, no. 11, pp. 1773–1782, 2005.

[Com02a] Comaniciu, D., and Meer, P., Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, no. 5, pp. 603–619, 2002.

[Fel04a] Felzenszwalb, P.F., and Huttenlocher, D.P., Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, 59, no. 2, pp. 167–181, 2004.

[Gon11a] Gong, H., and Shi, J., Conditional entropies as over-segmentation and under-segmentation metrics for multi-part image segmentation. Technical Reports (CIS), 2011.

[Lev09a] Levinshtein, A., Stere, A., Kutulakos, K., et al., TurboPixels: Fast Superpixels Using Geometric Flows. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31, no. 12, pp. 2290–2297, 2009.

[Mar01a] Martin, D., Fowlkes, C., Tal, D., et al., A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings, 2001, vol. 2, pp. 416–423 vol.2.

[Meiqq07a] Meila, M., Comparing clusterings - an information based distance. Journal of Multivariate Analysis, 98, no. 5, pp. 873–895, 2007.

[Pen13a] Peng, B., Zhang, L., and Zhang, D., A survey of graph theoretical approaches to image segmentation. Pattern Recognition, 46, no. 3, pp. 1020–1038, 2013.

[Pro12a] Protzel, P., and Neubert, P., Superpixel Benchmark and Comparison. Proc. of Forum Bildverarbeitung, 2012.

[Unn07a] Unnikrishnan, R., Pantofaru, C., and Hebert, M., Toward Objective Evaluation of Image Segmentation Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, no. 6, pp. 929–944, 2007.

8. APPENDIX

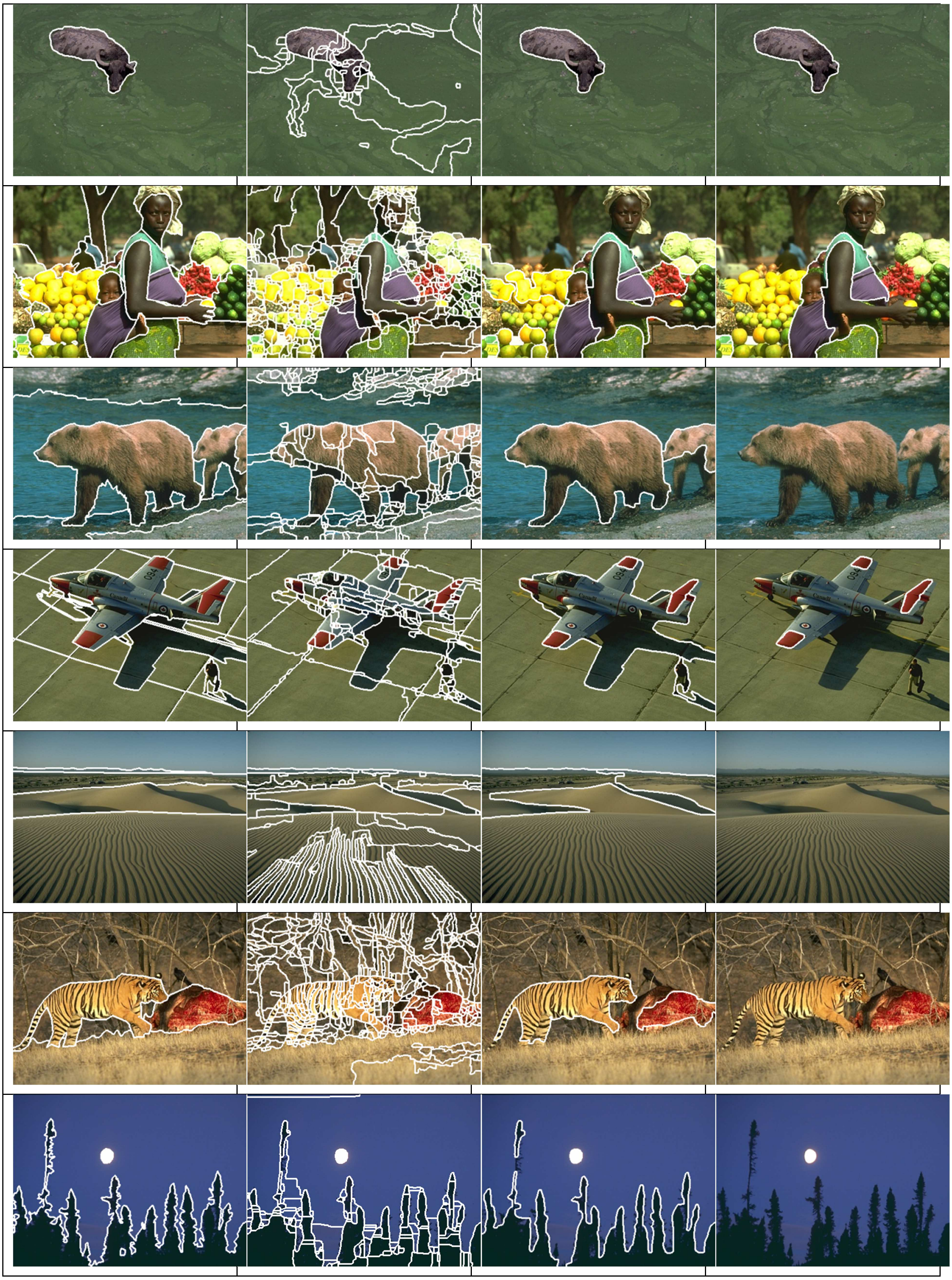


Figure 10. OWT-UCM segmentations at 0.05, 0.5 and 0.9, left to right. Reference images in first column

Interactive Tool and Database for Optic Disc and Cup Segmentation of Stereo and Monocular Retinal Fundus Images

F. Fumero[1], J. Sigut[2], S. Alayón[3]
University of La Laguna
Department of Computer Science
Faculty of Physics
Spain (38200), La Laguna, Tenerife
[1]ffumerob@ull.es, [2]sigut@isaatc.ull.es,
[3]silvia@isaatc.ull.es

M. González-Hernández[4], M. González de la Rosa[5]
University of La Laguna
Department of Ophthalmology
Hospital Universitario de Canarias
Spain (38320), La Laguna, Tenerife
[4]mgdelarosa@telefonica.net,
[5]martaglezhdez@gmail.com

ABSTRACT

Glaucoma is one of the leading causes of irreversible blindness in the world. Early detection is essential to delay its evolution and avoid vision loss. For this purpose, retinal fundus images can be used to assess the cup-to-disc ratio, the main indicator of glaucoma. Several automatic methods have been developed to compute this indicator, but the lack of ground truth of the optic disc and cup is an obstacle to evaluate and compare their results. In order to support clinicians to perform this task, an interactive tool for the segmentation of the disc and cup on stereo and monocular retinal fundus images has been developed. By using this tool, we have also built a new database of 159 stereo fundus images with two ground truth of disc and cup. The application and the database are both publicly available online. This work can serve as a learning environment for clinicians, as well as to evaluate the results of automatic segmentation algorithms.

Keywords
retinal fundus images, optic disc, optic cup, database, interactive segmentation, stereo

1 INTRODUCTION

A retinal fundus image is an image of the retina taken by a specialized camera, called fundus camera. This image modality plays an important role on diagnose eye diseases such as diabetic retinopathy or glaucoma.

Glaucoma is the leading cause of irreversible blindness in the population of industrialized countries [But12][Ala13] and refers to a group of diseases that affect the optic nerve and involves a loss of retinal ganglion cells. Early detection is crucial to prescribe appropriate treatment, in order to delay its evolution and avoid vision loss. The best known prevention methods are the regular assessment of the morphology of the optic nerve head (ONH), establishing the thickness of the fiber layer of the retinal optic nerve, and the subjective analysis of retinal sensitivity in the visual field of the patient.

The morphology of the surface of the optic nerve is controlled through direct observation methods, such as confocal laser scanning (HRT) or retinal fundus images. The latter, either monocular or stereoscopic, is the only modality which preserves the color and most pathologies of the retina. The main indicator of glaucoma is the cup-to-disc ratio (CDR) [Zha10][Tru13], which is the ratio of the size of the optic cup to that of the optic disc. Glaucoma can cause the cup to enlarge because of the loss of nerve fibers, leading to high values of the CDR [Bha10].

Figure 1 shows examples of retinal fundus images of a healthy and a glaucomatous eye. The images have been cropped to highlight the optic nerve head of the retina, also called the optic disc, which is the brightest part with elliptical shape. In both images, the contours of the optic disc and the optic cup have been superimposed in green and white, respectively. It can be seen that the cup-to-disc ratio is larger in the subject with glaucoma than in the healthy subject. As the cup-to-disc ratio is considered the main indicator of glaucoma, the correct detection of the optic disc and cup is a key factor for the early detection and treatment.

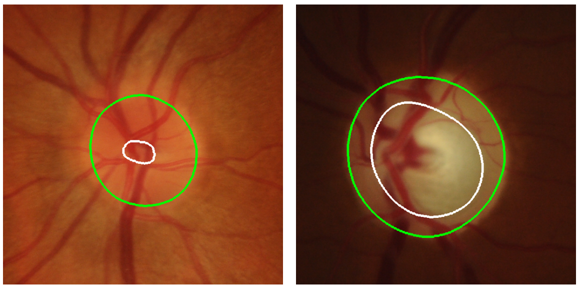


Figure 1: Examples of retinal fundus images of a healthy (left) and glaucomatous (right) subject, cropped to highlight the optic nerve head. The contours of the optic disc (green) and cup (white) have been superimposed.

There are numerous contributions aiming to automatically measure the CDR [But12][Jos10] but given the lack of totally objective instruments to obtain ground truth segmentations of the optic disc and cup, the only way to evaluate the results of these methods is to rely on expert segmentations [Tru13]. One of the obstacles is that most information remains locally in hospitals or research centers [Zha10] and only one public dataset for glaucoma diagnosis is available [Siv14], as far as we know.

The manual delimitation of the optic disc and cup is a difficult task, due to the high inter-patient variability of the ONH appearance [Ala13]. Hence, considerable experience is required to perform this task, but as some authors have suggested clinicians are not trained to use software to annotate images and create reference segmentations [Tru13]. To this end, it would be desirable to have intuitive annotation tools that can also be used to introduce clinicians to this practice.

In order to achieve this, we have developed an easy-to-use interactive segmentation tool, called DCSeg, specifically designed to manually segment the optic disc and cup. Its main features are the possibility of segmenting the cup as well as the disc, the ability to show an stereo image to ease the segmentation of the cup [Sto10] and a self-assessment mode to train or improve the abilities of a new user when creating ground truth of the disc and the cup.

We have used this tool to build a new database of stereo fundus images, with two reference segmentations of the optic disc and cup, created by two experts in ophthalmology from Hospital Universitario de Canarias. This database can be used in the self-assessment mode of DCSeg and for the evaluation of the results of automatic algorithms.

Both the segmentation tool and the database are publicly available online at the website of the Medical Image Analysis Group of the University of La Laguna (`http://medimrg.webs.ull.es/`).

The detailed description of the DCSeg software (section 3) and the database of stereo fundus images with ground truth of the optic disc and cup (section 4) constitute the core of this paper.

2 RELATED WORK

Firstly, we will analyze some tools for interactive or semiautomatic segmentation of medical images and, in particular, fundus images. Secondly, we will discuss some databases of retinal images that are available online.

2.1 Segmentation tools

The tools that allow labeling of medical images range from general-purpose image processing applications, such as the GIMP, to more specific ones, commercial and free, that allow the user to segment an image interactively. In this review, we will focus on specific tools freely available on the Internet, with a special emphasis on those designed to annotate the structures of the retina, like the optic disc and cup.

Most interactive segmentation tools for medical images work with 3D gray-scale images which is not useful in our case. Some applications that allow working with 2D color medical images are Ratsnake [Iak14], which uses active contours and has been successfully applied to different medical imaging domains, MRSeg [Fum13], an adaptable tool that can use any automatic segmentation algorithm to create regions that the user will later fuse or split to create the ground truth, or ilastik [Som11], an interactive segmentation environment with a trainable classifier based on Random Forest.

However, segmentation tools specifically designed for retinal fundus images are scarce or difficult to access. The most notable ones are the applications recently included in the VAMPIRE project [Per11]. In the context of this project, a semi-automatic tool to annotate the optic disc, fovea, junctions and vessel widths has been developed. Separate tools to manually segment the macula, the vessels and the optic disc [Gia11] have also been published. The latter allows the user to specify 10 points along the contour of the optic disc and then performs the fitting of an ellipse to generate the final segmentation. However, a serious disadvantage of this tool is that the fitted ellipse cannot be deformed to capture the irregular shapes of some discs. Moreover, the tool does not enable the user to annotate the optic cup or work with stereo images. DCSeg provides all these features, among others.

2.2 Retinal databases

There are some well-known databases of retinal fundus images that have been widely cited in the scientific literature, such as the DRIVE [Sta04], a database for vessel extraction, and the STARE project [Hoo03], created to validate the location of the optic nerve in retinal images.

Concerning databases that contain ground truth of the optic disc, we have found DiaRetDB1 [Kal], with 89 color fundus images annotated with different diabetic retinopathies and with some elliptical optic disc segmentations, and DRIONS-DB [Car08], which contains 110 images and 2 ground truths of the optic disc for each one. Some other examples are ARIA [Zhe12], RIM-ONE [Fum11] and MESSIDOR [Tec]. ARIA only provides optic disc segmentation for some of the images. RIM-ONE is a database specifically designed for optic disc segmentation of color fundus images. Up to date, it has 2 releases, the first one with 5 manual expert segmentation and 169 images classified into normal subjects and different glaucoma states; the most recent release with one expert segmentation of 455 images, split into healthy and glaucoma subjects. MESSIDOR consists of 1200 images with annotations of the optic disc by a single clinician of the University of Huelva [GA13].

There seems to be only one publicly available database with optic disc and cup reference segmentations, Drishti-GS [Siv14]. It consists of 101 monocular fundus images, divided into 50 training and 51 testing images, with four expert segmentations of the disc and cup for the training set. Some authors have published papers describing similar databases, such as ORIGA-light [Zha10] or, more recently, ACHIKO-K [Zha13], but they cannot be easily accessed. ORIGA-light contains 650 fundus images annotated by one clinician using an ellipse fitting. ACHIKO-K consists of 258 images from glaucoma patients, with expert segmentations generated via polynomial spline fitting.

As far as we know, and as pointed out in the exhaustive and extensive revision carried out in [Tru13], there are no publicly available stereo datasets for glaucoma diagnosis and validation. Therefore, to the best of our knowledge the proposed database seems to be the first publicly available database of stereo fundus images with ground truth of the optic disc and cup.

3 INTERACTIVE SEGMENTATION TOOL

The main aims of this work are to ease the creation of data sets of reference segmentations of the optic disc and cup and to train users to perform this task.

For this purpose, we have developed DCSeg, a desktop application with a graphical user interface for the interactive segmentation of the optic disc and cup in retinal fundus images. It allows to manually segment the optic disc and the cup separately, using monocular or stereo fundus images. The tool has been designed to be able to create several reference segmentations of the same image, by one or more experts. Moreover, it can be used to train a new user on the segmentation of this kind of structures through the self-assessment mode of the application. By using this feature, the tool compares the segmentation of the user to the gold standard of the same image developed by the experts, thus helping the user to improve their skills.

This application has been developed entirely in Java and has been tested on Java SE 6, 7 and 8, so it could be used in the most popular operating systems.

In the following subsections we will describe the features of the tool in detail.

3.1 Segmentation of monocular fundus images

By using monocular fundus images the user can manually segment either the disc, the cup or both. As we previously pointed out, some other tools [Gia11][Zha10] require to select some points of the image to perform an ellipse fitting based on least square fitting algorithms. Sometimes, it is difficult to select these points to achieve the desired fitting because the fitted ellipse is only shown once the user has selected all the points. Besides, the result is an ellipse that cannot be deformed to capture individual variability.

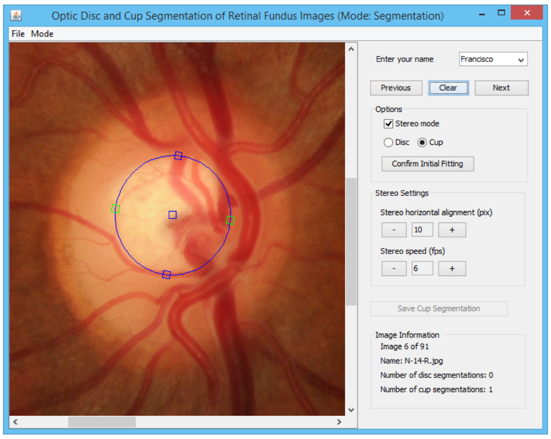
In order to ease the process of manual segmentation, a different strategy has been implemented in DCSeg. Firstly, it allows to make an initial segmentation of the contour of the optic disc and cup by adjusting an ellipse on top of the image. This initial approximation can be easily achieved by using the green and blue controls showed in Fig. 2 to adjust the position and the shape of the ellipse.

Secondly, after this initial fitting, the ellipse can be optionally deformed to capture the irregular shape of some structures. In order to do this, the application gives the option to refine the segmentation by the adjustment of 16 predefined radii, as showed in Fig. 3.

The application can also be used by more than one expert to create several reference segmentations of the same image. The number of disc and cup segmentations of the same image are clearly showed in the application interface. This feature helps to evaluate the intra- and inter-observer variability and to generate richer and more reliable gold standards.

3.2 Usage of stereo fundus images

Alternatively, stereo fundus images can be used to segment the cup in an easier way. The stereo output format of the Kowa WX 3D fundus camera has been adopted as the input format in DCSeg. An example of this format is shown in Fig. 4. It consists of two photographs of the same eye taken at the same time from slightly


Optic Disc and Cup Segmentation of Retinal Fundus Images (Mode: Segmentation)

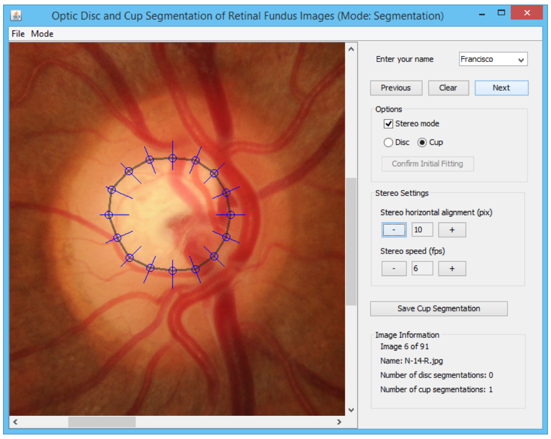
Enter your name: Francisco

Options
☑ Stereo mode
○ Disc ● Cup

Stereo Settings
Stereo horizontal alignment (pix): 10
Stereo speed (fps): 6

Image Information
Image 6 of 91
Name: N-14-R.jpg
Number of disc segmentations: 0
Number of cup segmentations: 1

Figure 2: Approximation of the optic cup by an ellipse.


Optic Disc and Cup Segmentation of Retinal Fundus Images (Mode: Segmentation)

Enter your name: Francisco

Options
☑ Stereo mode
○ Disc ● Cup

Stereo Settings
Stereo horizontal alignment (pix): 10
Stereo speed (fps): 6

Image Information
Image 6 of 91
Name: N-14-R.jpg
Number of disc segmentations: 0
Number of cup segmentations: 1

Figure 3: Refinement of the initial fitting of the cup by adjusting the predefined radii.

different angles and showed together on one image to form a stereo pair.

The *Stereo mode* of DCSeg simulates 3D effect by looping the two images of the stereo pair, one on top of the other, as the frames of an animated image, which is often known as *animated stereograph*. The frequency of the animation is 6 frames per second by default.

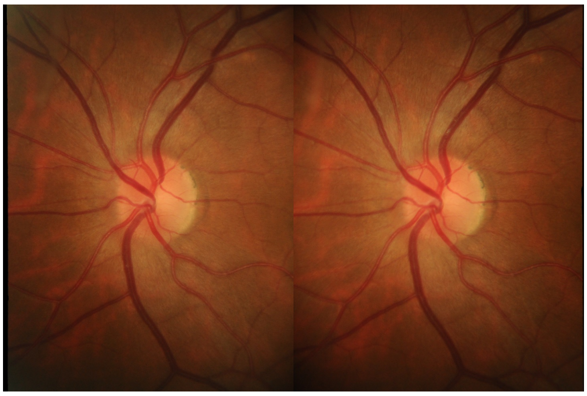


Figure 4: Example of the stereo image format accepted by DCSeg. The ONH of a subject is captured from two slightly different angles at the same time using a stereoscopic fundus camera.

As the two images are taken from different angles, this technique allows the user to recover part of the 3D scene, which provides some depth information of the cup thus easing its segmentation. This avoids the use of a stereoscope or a special stereo viewer but still takes advantage of the utility of stereo images to evaluate the optic disc [Sto10].

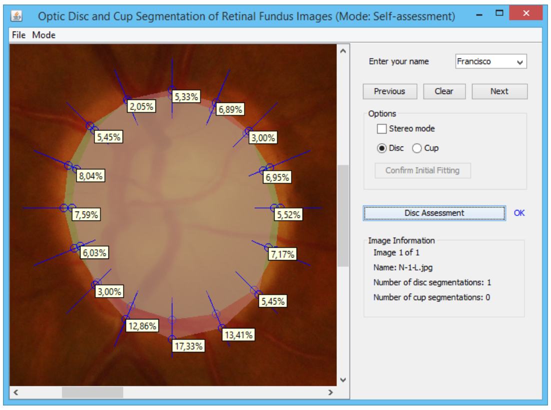
The application provides some controls to the user in order to deal with the *Stereo mode*: the *Stereo horizontal alignment* and the *Stereo speed* (Fig. 2, 3). The former, *Stereo horizontal alignment*, allows the user to properly align the left and right images of the stereo pair, as they are vertically aligned but not necessarily horizontally aligned. A proper horizontal alignment is necessary to see the 3D effect of the animation.

The latter, *Stereo speed*, gives the user the ability to change the frames per second, increasing or decreasing the speed of the animation.

3.3 Learning and self-assessment mode

As we previously stated and according to [Tru13], annotating the structures of a medical image using a computer application is a task that clinicians are not normally trained for.

In order to introduce ophthalmology experts into this practice, a self-assessment mode has been implemented. This mode of the application can be used to learn how to segment both the disc and the cup. It allows the user to perform a segmentation as described in the previous sections and compare the result to the ground truth of the image being segmented, showing the variability percentage of each one of the radii (Fig. 5).


Optic Disc and Cup Segmentation of Retinal Fundus Images (Mode: Self-assessment)

Enter your name: Francisco

Options
□ Stereo mode
● Disc ○ Cup

Disc Assessment OK

Image Information
Image 1 of 1
Name: N-1-L.jpg
Number of disc segmentations: 1
Number of cup segmentations: 0

Figure 5: Self-assessment mode of DCSeg showing the variability percentage of all the radii between the ground truth and the user segmentation of the disc.

This feature is also useful to visually evaluate if two experts have different criteria segmenting both structures, to assess the intra-observer variability and to detect errors in the database.

4 DATABASE OF STEREO FUNDUS IMAGES

A new online database of stereo retinal fundus images for glaucoma diagnosis and validation has been developed. This is the third release of the Retinal IMage database for Optic Nerve Evaluation, RIM-ONE [Fum11], and consists of 159 stereo fundus images and 2 expert reference segmentations of optic disc and cup.

The previous releases of RIM-ONE were focused on the segmentation of the optic disc, while this new one also adds optic cup ground truths as well as stereo images. The expert segmentations have been carried out by using the proposed tool DCSeg. The database and the tool can be both used for research and educational purposes, without requesting permission to the authors.

The rest of this section describes the dataset, the ground truth and the evaluation of the results.

4.1 Dataset

The database consists of 159 images collected at the Hospital Universitario de Canarias and divided by the experts into two groups: one group for healthy subjects and the other group for glaucoma or suspicion of glaucoma patients. The group of healthy subjects consists of 85 images (46 females and 39 males, 48 ± 15 years) and the other group contains 39 confirmed glaucomas (17 females and 22 males, 68 ± 11 years) and 35 glaucoma suspects (21 females and 14 males, 62 ± 12.1 years).

The study was performed in accordance with the ethical standards established in the 1964 Helsinki declaration. Ethical committee approval was obtained and the patients were informed about the objectives of this study. All of the studied subjects were selected arbitrarily and do not belong to longitudinal cases.

All the images were taken by a non-mydriatic Kowa WX 3D stereo fundus camera, with specific flash intensities to avoid color saturation. They are centered on the ONH with a field-of-view of 34^{o} and the stereoscopic images are captured in the same camera shot, giving a resolution of 2144 × 1424 pixels in the format showed in Fig. 4. The image files of the dataset are named as *Category-K-EyeSide.jpg*, where *Category* can be Normal (N), Glaucoma (G) or Suspect (S), *K* is a number assigned to each image and *EyeSide* can be Left (L) or Right (R).

4.2 Ground truth description

The interactive tool DCSeg has been used in *Stereo mode* to carry out the ground truth of the database. As a result, some text files were generated for each expert segmentation. Then the files were processed using some Matlab scripts, which are also publicly available, to obtain an average segmentation and an interpolated mask. The format and content of each type of ground truth file are described below.

- DCSeg segmentation file

  This file contains the parameters of the ellipse used for the initial segmentation of the disc and cup, the coordinates of each one of the points over the predefined radii, as well as the coordinates of the center of the radii and the ellipse. The name of this file is of the form *Category-K-EyeSide-M-Region-Expert.txt*, where the first 3 elements identify the image, *Region* refers to *Disc* or *Cup*, *Expert* is the expert who performed the segmentation, and *M* is an order number to properly store different segmentations of the same image and region by the same expert.

- Average segmentation file

  All of the text files containing an expert segmentation of the same image and region were fused together into a new file. This was done by a Matlab script that averages the points of the same radius (Fig. 3) from different segmentations, creating an average segmentation file. This file is named as *Category-K-EyeSide-Region-Avg.txt* and contains the average coordinates of the center and the points.

- Interpolated mask file

  Another Matlab script was used to interpolate the points of the radii and create a smoothed mask from the text files described previously. The interpolated mask is stored as a binary image in PNG and MAT formats, for the experts and the average segmentation using the same name convention as before. Figure 1 shows examples of two images of the dataset where the contours of the interpolated masks for one of the experts have been superimposed.

4.3 Evaluation of the database

The expert segmentations of the database have been evaluated using the Jaccard index, the variability measure defined in [Fum11] and the estimated vertical CDR.

The Jaccard similarity index between a segmentation S and a reference segmentation R is the size of the intersection divided by the union of the two segmentations (Eq. (1)).

$$J(S,R) = \frac{|S \cap R|}{|S \cup R|} \qquad (1)$$

The variability percentage (VP) defined in eq. (2) measures the variability between the boundary of a segmentation and a reference in 8 points of the contour every 45^{o}, covering the most important zones of the ONH. The terms $d_i(S)$ and $d_i(R)$ measures the distance from the center of the reference to the $i-th$ point of, respectively, the contour of S and R.

$$VP_i(S,R) = \frac{|d_i(S) - d_i(R)|}{d_iR} * 100, i \in [1..8] \quad (2)$$

The vertical CDR (VCDR) is the ratio between the vertical size of the cup to that of the disc, as shown in Eq. (3). The vertical size is the maximum size in the vertical direction of the corresponding structure [Bha10].

$$VCDR = \frac{Vertical\,Cup\,Size}{Vertical\,Disc\,Size} \quad (3)$$

In order to perform this evaluation, the healthy and the glaucoma group have been studied independently. For the Jaccard index and the VP, we have chosen the average segmentation described in section 4.2 as the reference R and each individual expert segmentation as S, evaluating the disc and the cup separately. Tables 1 and 2 contain the results of both measures, averaged for the two experts. The results show high similarity and low variability, suggesting an agreement between the experts and the reference. It also seems to be a stronger agreement on the disc segmentation than on the cup. In the segmentation of the disc, there are no significant differences between the healthy and the glaucoma group. However, the difference between the two groups when segmenting the cup is significant ($p < 0.001$, two-sample t-test), giving better values in the glaucoma group.

	Healthy	Glaucoma
Disc	0.96 ± 0.02	0.95 ± 0.03
Cup	0.79 ± 0.11	0.86 ± 0.10

Table 1: Average Jaccard similarity index ($M \pm SD$).

	Healthy	Glaucoma
Disc	2.31 ± 1.11	2.64 ± 1.79
Cup	12.98 ± 8.73	8.22 ± 6.47

Table 2: Average Variability Percentage ($M \pm SD$).

Table 3 shows the estimated VCDR from the expert segmentations and the correlation coefficient r between the VCDR of the two experts. It can be seen that the mean values of the VCDR are lower in the healthy group than in the glaucoma group, as it was expected. The correlation between the experts is high in both groups, but slightly higher in the glaucoma group, which is in accordance with the previous measures in the cup. Despite the results obtained for the healthy group, significant differences between the VCDR of the two groups have been found ($p < 0.0001$, two-sample t-test). This suggests that the VCDR can be estimated from the manual segmentations, obtained through the use of DCSeg.

	Expert 1	Expert 2	r
Healthy	0.41 ± 0.10	0.42 ± 0.11	0.74
Glaucoma	0.60 ± 0.17	0.60 ± 0.17	0.88

Table 3: Average Estimated VCDR ($M \pm SD$) and correlation coefficient r.

5 CONCLUSIONS

In this work, we have presented an interactive tool called DCSeg for the segmentation of the optic disc and cup in monocular and stereo fundus images which has been used to build a new database of stereo retinal images with their corresponding ground truth. The application also features a special mode to train clinicians to carry out this task. The database has been evaluated using several measures and we have found a good agreement between the experts, concluding that DCSeg is helpful to perform manual segmentations and estimate the cup-to-disc ratio.

As future work, our effort will be focused on the further development of the database, extending the number of ground truths with expert segmentations from any groups willing to collaborate with this project. For this purpose, a web version of DCSeg will be released in the next months. As the number of ground truth increases, a more advanced algorithm could be implemented to combine them into a unique reference segmentation, instead of using a simple average procedure. The usability of the DCSeg interface will also be improved based on user experience.

6 ACKNOWLEDGMENTS

This research has been supported by a Doctoral Scholarship from the Spanish Government (AP2012-6800), and has also been partially funded by the Project PI12/02301 from the Fondo de Investigación Sanitaria, Instituto Carlos III (co-financed with Feder funds). The authors acknowledge the funding granted to the University of La Laguna by the Canarian Agency for Research, Innovation and Information Society, financed in an 85% by the European Social Fund.

7 REFERENCES

[Ala13] Alayon, S., Gonzalez de la Rosa, M., Fumero, F.J., et al., Variability between experts in defining the edge and area of the optic nerve head. Archivos de la Sociedad Espanola de Oftalmologia (English Edition), 88, no. 5, pp. 168–173, 2013.

[Bha10] Bhartiya, S., Gadia, R., Sethi, H.S., et al., Clinical evaluation of optic nerve head in glaucoma. Journal of Current Glaucoma Practice, 4, pp. 115–132, 2010.

[But12] Buteikiene, D., Paunksnis, A., Barzdziukas, V., et al., Assessment of the optic nerve disc and excavation parameters of interactive and automated parameterization methods. Informatica, 23, no. 3, pp. 335–355, 2012.

[Car08] Carmona, E.J., Rincón, M., García-Feijoó, J., et al., Identification of the optic nerve head with genetic algorithms. Artificial Intelligence in Medicine, 43, no. 3, pp. 243–259, 2008.

[Fum13] Fumero Batista, F.J., Garcia Llarena, S.J., Nunez Regalado, O., et al., MRSeg - herramienta interactiva para generar segmentaciones de referencia de imagenes medicas, Terrassa2013.

[Fum11] Fumero, F., Alayon, S., Sanchez, J., et al., RIM-ONE: An open retinal image database for optic nerve evaluation. In 2011 24th International Symposium on Computer-Based Medical Systems (CBMS), 2011, pp. 1–6.

[GA13] Gegundez-Arias, M.E., Marin, D., Bravo, J.M., et al., Locating the fovea center position in digital fundus images using thresholding and feature extraction techniques. Computerized Medical Imaging and Graphics, 37, no. 5-6, pp. 386–393, 2013, http://www.uhu.es/retinopathy/muestras/Provided_Information.zip.

[Gia11] Giachetti, A., Chin, K., Trucco, E., et al., Multiresolution localization and segmentation of the optical disc in fundus images using inpainted background and vessel information. In 2011 18th IEEE International Conference on Image Processing (ICIP), 2011, pp. 2145–2148.

[Hoo03] Hoover, A., and Goldbaum, M., Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. IEEE Transactions on Medical Imaging, 22, no. 8, pp. 951–958, 2003.

[Iak14] Iakovidis, D.K., Goudas, T., Smailis, C., et al., Ratsnake: A versatile image annotation tool with application to computer-aided diagnosis. The Scientific World Journal, 2014, 2014.

[Jos10] Joshi, G., Sivaswamy, J., Karan, K., et al., Optic disk and cup boundary detection using regional information. In 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2010, pp. 948–951.

[Kal] Kalesnykiene, V., Kamarainen, J.k., Voutilainen, R., et al., DIARETDB1 diabetic retinopathy database and evaluation protocol.

[Per11] Perez-Rovira, A., MacGillivray, T., Trucco, E., et al., VAMPIRE: Vessel assessment and measurement platform for images of the REtina. International Conference of the IEEE Engineering in Medicine and Biology Society., 2011, pp. 3391–3394, 2011.

[Siv14] Sivaswamy, J., Krishnadas, S.R., Chakravarty, A., Joshi, G.D., Ujjwal, et al., A Comprehensive Retinal Image Dataset for the Assessment of Glaucoma from the Optic Nerve Head Analysis. JSM Biomed Imaging Data Pap 2(1): 1004, 2015.

[Som11] Sommer, C., Straehle, C., Koethe, U., et al., "ilastik: Interactive learning and segmentation toolkit". In 8th IEEE International Symposium on Biomedical Imaging (ISBI 2011), 2011.

[Sta04] Staal, J., Abramoff, M., Niemeijer, M., et al., Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging, 23, no. 4, pp. 501–509, 2004.

[Sto10] Stone, R.A., Ying, G.s., Pearson, D.J., et al., Utility of digital stereo images for optic disc evaluation. Investigative Ophthalmology & Visual Science, 51, no. 11, pp. 5667–5674, 2010.

[Tru13] Trucco, E., Ruggeri, A., Karnowski, T., et al., Validating retinal fundus image analysis algorithms: issues and a proposal. Investigative ophthalmology & visual science, 54, no. 5, pp. 3546–3559, 2013.

[Tec] Techno-Vision, P., Méthodes d'evaluation de systèmes de segmentation et d'indexation dédiées à l'ophtalmologie rétinienne. http://messidor.crihan.fr/.

[Zha10] Zhang, Z., Yin, F.S., Liu, J., et al., ORIGA-light: An online retinal fundus image database for glaucoma analysis and research. In International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2010, pp. 3065–3068.

[Zha13] Zhang, Z., Liu, J., Yin, F., et al., ACHIKO-K: Database of fundus images from glaucoma patients. In 2013 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013, pp. 228–231.

[Zhe12] Zheng, Y., Hijazi, M.H.A., and Coenen, F., Automated "disease/no disease" grading of age-related macular degeneration by an image mining approach. Investigative Ophthalmology & Visual Science, 53, no. 13, pp. 8310–8318, 2012.

Edge-aware Normal Estimation by Rotated Bilateral Sampling

Viktor Kovacs, Gabor Tevesz
Budapest University of Technology and Economics
Department of Automation and Applied Informatics
Magyar Tudosok krt. 2. QB207
1117 Budapest, Hungary
{viktor.kovacs},{gabor.tevesz}@aut.bme.hu

ABSTRACT

In this paper we deal with edge preserving surface normal estimation and crease edge detection in discretized range images. Such range images consist of few discrete quantization levels due to the data acquisition method (short base distance stereo), or when the distance variation of the examined surface is low, compared to the disparity quantization levels. We propose a method for normal estimation and crease edge detection using iso-range curves and rotated bilateral filter based sampling. Iso-range curves are used to extract sparse, but reliable range image points. Samples are first selected by a rotated weight matrix and a plane is fitted on such samples. Simple statistics are gathered during the rotation of the weight matrix, in order to find the best fitting plane and extract crease edge measure. Such information may be used for further range image processing: segmentation, mapping, localization, object detection, recognition etc. Results are shown for both synthetic and real range images. It was shown that applying the proposed method resulted in more accurate normal estimations, crease edges were not smoothed and crease edges were successfully detected.

Keywords
range image, normal estimation, edge detection, plane fitting, short base distance

1 INTRODUCTION

The apparatus for acquisition and processing of 3D geometry data became affordable and compact thus appearing in a wide range of applications (mobile robotics, photogrammetry etc.). Several methods and implementations are available for 3D sensing, each balancing with different features. The optimal must be chosen for each application (cost, precision, range etc).

Time of flight (ToF) based methods provide the most accurate results even at long range at high costs. Conventional stereo is widely used to reconstruct 3D geometric data due to the low development price. Stereo triangulation is based on disparity estimation between the two viewpoints. Surface texture, geometry and lighting affects the disparity estimation and thus the reconstructed geometry. Feature points may be used for more accurate matching between viewpoints but it results in a sparse disparity map. In case of homogeneous texture it might be impossible to estimate the geometry. For such reasons structured light based methods are also used: one camera is substituted by a calibrated projector that emits a known or a series of known light patterns thus homogeneous surfaces can be textured in this way. Disadvantages are short range and non-passive operation: it needs to emit enough light that could be detected. In some cases active operation is not admissible.

Stereo camera based methods provide a dense disparity map based on similarity (normalized cross correlation, sum of differences etc). Dense maps may be transformed to a range image by associating a range value based on the disparity measure and known optical properties. Usually these depth maps are stored as range images, where pixel intensity encodes the depth (Z) coordinate value in order to keep beneficial properties of such images: regular sampling, vicinity information, simple surface and triangle mesh generation etc.

In this paper we deal with range images to utilize specific features and errors associated with structured light based image acquisition methods. The research aims to provide a set of methods to handle range images that were acquired using short baseline distance and the disparity map was estimated using traditional (SAD: sum of absolute differences or NCC: normalized cross correlation) methods. Due to the short baseline distance disparity values are also small and quantized in image space. Such quantization leads to the discretization of range values as well:

$$z_i = Bf\frac{1}{d_i}, \tag{1}$$

where z_i is the calculated depth value from disparity d_i using B baseline distance and f focal length. When the disparity map was quantized (to pixels) the reconstructed depth map would be quantized as well, and would be inversely proportional to the disparity. Range images $Z(u,v)$, $0 \leq u < n, 0 \leq v < m$ suffering from strong quantization consist of few discrete range values:

$$|\{Z(u,v)\}| << nm. \tag{2}$$

Such strong quantization noise cannot be considered random, and has significant effects on algorithms used for low level range image processing. Hough transform or RANSAC based model (ie. plane) fitting may easily find better but improper explanations of surface parts described by discrete range values. Such false surface regions show as quantization levels, planar patches perpendicular to the z axis.

A framework is presented for processing such layered range images, specifically edge-aware normal estimation and edge detection. Our method consists of two major steps: a preprocessing step, where low level features are extracted first and postprocessing, where such features are utilized for further analysis.

The preprocessing step involves layer separation, filtering, skeleton extraction. First the quantized range image is broken into binary images, each image describing a layer. Each binary layer is filtered in order to reduce noise, finally skeletons are extracted using thinning to describe the layers. It is assumed that such skeletons estimate iso-range curves on surfaces where the quantization error is minimal. Due to perspective projection such centerline estimation in image space is biased. However our results show that such error is not significant in practical cases, only at extreme cases where the surface is steep and the quantization step is significantly large.

During postprocessing such skeletons are used in plane, edge and corner detection. Such features can be used in registration problems, mapping, localization, or object detection.

This paper focuses on normal estimation in such sparse range image point sets while keeping both jump and crease edges. Surface normals provide low level features used in subsequent processing steps in range image understanding. The naive approach of local surface estimation by plane fitting on a local neighborhood of pixels lead to significant errors in discretized range images. Based on the layer widths (surface orientation and quantization), the local neighborhood size and the weighting, naive estimation would give significantly different results. Using a small neighborhood for sampling would mostly result in sampling from one layer, thus providing a normal parallel to the depth direction. Near layer edges a perturbation would be observed of the incorrectly estimated surface normal. In order to improve estimation, data uncertainty must be estimated. As quantization is not random, spatial information may be introduced for uncertainty estimation. In the proposed method as a simplification, it is assumed that the centerlines of layers carry reliable depth information, these shall be used for model fitting, other layer pixels are ignored.

In this paper the rotated bilateral sampling method is proposed, by which edge aware fitting of models, in the given example local planar segments for normal estimation are possible. With side information of edges, the estimation process may be sped up.

The paper is organized as follows: in section 2 related work is presented, in section 3 the proposed algorithm is shown. Results of simulation and real images are presented in section 4, finally in section 5 results are discussed and conclusions are drawn.

2 RELATED WORK

Bilateral filtering was introduced in [17]. Such filters combine closeness (spatial) and similarity (value) filtering in one general filter:

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{f}(\xi)c(\xi,\mathbf{x})s(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))d\xi \tag{3}$$

$$k_r(\mathbf{x}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(\xi,\mathbf{x})s(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))d\xi \tag{4}$$

where $\mathbf{f}(\mathbf{x})$ denote the input image, $\mathbf{h}(\mathbf{x})$ the filtered output. Functions $c(\xi,\mathbf{x})$ and $s(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))$ define the closeness and similarity functions. Weights $k(\mathbf{x})$ are applied in order to preserve units.

On smooth regions where the variance of the values is low, it acts as a standard domain filter. On regions containing a sharp edge, where values differ significantly, values would be taken only from samples similar to the center value. As the kernel is not spatially invariant and is based on the original image contents, FFT and other methods are not applicable to speed up calculation.

Such filters are exceptionally popular for range image processing as object boundaries are not blurred with background information. Typical closeness and similarity filters are Gaussian. In the proposed method the filters are also Gaussian.

A hybrid solution with weighted median filtering is presented in [18] for range image upscaling using high resolution intensity images. Trilateral filters also take the gradient into consideration, in [14] such filters are used for upscaling range images.

Bilateral filters were also proposed as edge detectors in [8]. A high-pass closeness (domain) kernel is combined with an inverted Gaussian similarity (range) kernel.

Several types of edges can be differentiated in range images. Step or jump edges show the most resemblance to intensity image counterparts, they appear as depth discontinuities at object boundaries. Crease edges show as a significant change in normal direction. Smooth edges are identified by abrupt change of the surface curvature while the normal changes gradually. Most papers do not deal with smooth edges. According to the data acquisition circumstances, false edges may appear around regions of unknown depth values: using stereo triangulation occlusion may happen or distances may be larger than what the rangefinder can handle. Such edges must be omitted or handled accordingly. If layered range images were handled as intensity images an other edge type could be identified. These edges appear between quantization levels but do not represent any type of real edge. Such edges are usually noisy thus cannot be utilized directly, this is a reason skeletons were introduced instead of layer edges.

In [4] the bilateral grid was introduced for edge-aware algorithms. The method involves transforming the image to a higher dimensional grid along the similarity axis, such that pixels representing different patches (separated by an edge) are grouped into different grid cells. This method can also be used to re-express the bilateral filtering problem as a linear filter in a higher dimensional space. Sampling rate over the spatial domain controls the smoothing, sampling rate of the range axis defines the degree of edge preservation. Bilateral filtering can be expressed as 3D convolution between grid cells. The division by the weights are delayed, and data are represented by a variety of homogeneous coordinates. The dual operation of the grid generation is division (by the homogeneous coordinate) and slicing. Our representation is a specialized form of the bilateral grid, where the spatial sampling rates are 1, the range sampling corresponds to the available range layers, thus one range pixel is associated to each grid cell.

Surface normals provide basic features for higher level range image understanding such as segmentation, mapping, navigation, object recognition or detection. Authors of [9] compare several methods (different variations of singular value decomposition and principal component analysis, triangle based averaging) for surface normal estimation evaluating the tradeoff between precision and speed. Joint surface and surface normal reconstruction is shown in [19] using statistical methods for improved robustness. In [1] normal estimation is optimized for reduced computational demand transforming range images to spherical coordinate system, giving spherical range images. Normals may be directly extracted from such representation. In [13] directional joint bilateral filters are introduced to take edge direction into account during filtering. State of the art methods [6] involve integral images for surface normal estimation in point clouds. Integral images simplify summing over a rectangular region as only the values at the corners of the rectangle is needed.

The sampling matrix we proposed shows some resemblance to the one used in the Kuwahara filter and it's modifications [12]. The original filter uses four square regions around the sampled point where mean of the subregion is applied to the center pixel where the standard deviation is the lowest. Generalizations involve rotated circular and elliptical filter kernels.

Several methods are used for fitting planar surfaces to sample data. Hough-trasformation may be extended to 3D for plane detection [7, 2]. As the accumulator space has a higher dimensionality accumulator space design must be made carefully [3]. Model fitting in noisy data and high number of outliers are usually done by using a variation of the RANSAC algorithm [5]. Hybrid methods were also developed to fuse advantages of the methods. In [16] both Hough and RANSAC based methods are used at multiple resolutions.

3 PROPOSED METHOD

In this section we present a new method for local surface normal estimation in discretized range images. Estimated normals can be used for further analysis of the images, such as segmentation [11], edge detection [10], smoothing etc. while it also provides basic information for higher level semantic analysis such as object or landmark recognition.

3.1 Skeletonization

First the layered range image is broken into binary images representing each quantization value. Next these binary images are filtered in order to reduce noise that might be present near the edges of layers. Small detached patches are removed from the layers using the connected components algorithm. Morphological operations are applied in the resulting binary image: dilation and erosion in order to smooth the transitions between layers.

Next a thinning algorithm is applied for skeleton extractions. Skeletons are one pixel width lines that represent the centerline of binary images. There are numerous thinning algorithms, [15] was implemented, that produces few side branches. The resulting skeletons are broken into skeleton segments: such segments consist of skeleton pixels between junctions or endpoints. Skeletons are pruned by removing remaining unwanted short segments.

In our framework we utilize these iso-range skeletons for further processing. It is assumed that these iso-range lines describe surface crossections that have minimal quantization noise on linear surfaces. Centerline estimation error due to projection is minimal in practical cases. Figure 1 shows the iso-range skeletons of a sample simulated scene.

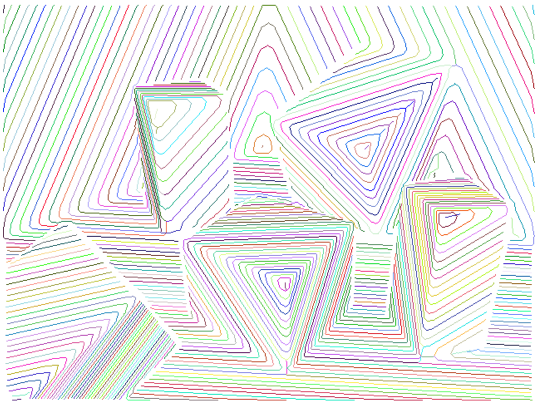


Figure 1: *Iso-range skeletons extracted from each layer of the discretized range image.*

3.2 Surface normal estimation

Normal estimation is one of the most fundamental steps in range image processing. Normals can be used for plane segmentation, edge detection etc.

Previously we proposed a method [10] for surface normal estimation by a variation of the forward difference method but adapted for 3D iso-range curves. The method can also be extended to utilize multiple layer information to estimate the gradient. Figure 2 illustrates the process.

First the skeleton tangential orientation v is estimated by sampling $n_{FitLine}$ number of skeleton points in image space. The total least squares method is applied to find the best fitting direction: the eigenvector related to the larger eigenvalue of the corvariance matrix is evaluated. This tangential direction is also the tangential of the skeleton in 3D, and it specifies a plane P. In order to estimate the binormal $\mathbf{p}^*$ must be identified on an adjacent layer (l_{i+1}), where the skeleton of the adjacent layer crosses P plane. By knowing the tangential and the binormal, the surface normal $\mathbf{n} = \mathbf{b} \times \mathbf{v}$ can be calculated. By identifying several binormals, not only on one adjacent layer, but on several, the normal estimation may be improved for planar regions, but near edges normals may be smoothed. By default this method identified one binormal on the following layer thus being noise sensitive. Estimation failed or was ignored near crease edges, due to the significant change in layer skeleton orientation.

In this paper we present a method that overcomes such problem, providing edge preserving surface normal estimation, using a rotated bilateral filter kernel for sampling and principal component analysis (PCA). PCA is an orthogonal transformation, which maximizes the variance of the samples along the principal directions. The eigenvectors and eigenvalues of the covariance matrix define the principal directions and components.

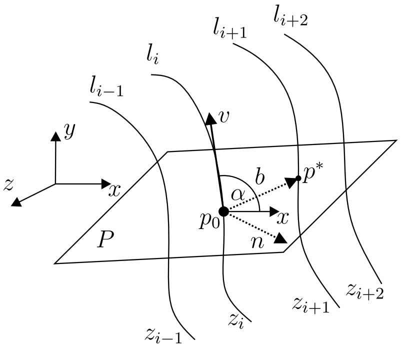


Figure 2: l_i: *layer skeletons,* v *tangent,* b *binormal,* n *normal direction*

First a set of sampling matrices are constructed as a function of θ and N.

$$\mathbf{S}_N(\theta)_{i,j} = \begin{cases} e^{-(a^2+b^2)} & \text{if } 0 \le \phi \le \pi \\ 0 & \text{otherwise} \end{cases} \tag{5}$$
$$i,j \in 1..(2N+1)$$

where

$$a = \frac{\sin(\phi)R}{Ns_2}, \quad b = \frac{\cos(\phi)R}{Ns_1}, \tag{6}$$

$$R = \sqrt{(i-N-1)^2 + (j-N-1)^2}, \tag{7}$$

$$\phi = \text{atan2}(j-N-1, i-N-1) + \pi + \theta. \tag{8}$$

s_1 and s_2 modify the shape and weight falloff of the sampled pixels. In our implementation we set $\theta = i/18\pi$, where $i = 0..35$. The constructed sampling matrices are illustrated in Figure 3. $\mathbf{S}_N(\theta)$ matrices are calculated only once and stored in a look-up table. To modify the behavior around edges or corners the opening angle may be changed by modifying the π constant in (eq. 5).

Such matrices are used for sampling skeleton points in image space for plane fitting. For each skeleton point a sampling matrix is selected by taking skeleton distance into account. Skeleton distance $d_s(\mathbf{p})$ is given during skeletonization by the number of steps after the centerline point is reached:

$$N(\mathbf{p}) = \max(N_{min}, \min(N_{max}, [s_N d_s(\mathbf{p})])) \tag{9}$$

where N_{min} and N_{max} denote the minimal and maximal size parameter of the sampling matrices, s_N denote the scale multiplier for selecting the size. Such selection of the sampling size enables better adaptation to skeleton (surface information) density.

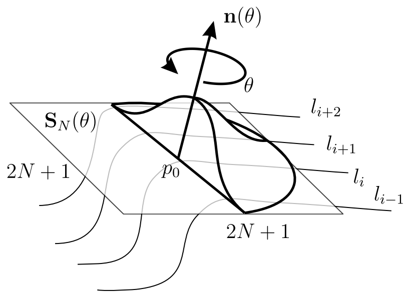


Figure 3: *The constructed sampling matrices are similar to folded bivariate Gaussian distribution pdfs, but rotated by θ around the center point. It serves as defining weights for sampled skeleton pixels but the weights are also modified by a bilateral filter. l_i denote layer skeletons.*

In order to estimate the local surface normal we first estimate the covariance matrix of the sampled skeleton points.

$$\Sigma(\theta) = \frac{0.5\sum_{k=1}^{N} w_k}{\left(\sum_{k=1}^{N} w_k\right)^2 - \sum_{k=1}^{N} w_k^2} \sum_{k=1}^{N} w_k \left(\mathbf{p}_k - \mathbf{p}_0\right)^T \left(\mathbf{p}_k - \mathbf{p}_0\right) \tag{10}$$

where $\mathbf{p}_0$ is the selected center point, $\mathbf{p}_k$ are the sampled skeleton points around the center in N radius. The estimation of the cross-variances relative to $\mathbf{p}_0$ means that the algorithm assumes that there are points mirrored to $\mathbf{p}_0$ on the zero side of $\mathbf{S}(\theta)$. To accept the samples N_s number of points must be sampled from at least N_l number of layers.

Weights w_k are given by the appropriate element of the sampling matrix and a scaled difference in depth (Z):

$$w_k = \mathbf{S}(\theta)_{i,j} \cdot \exp\left(-\frac{(Z(p_k) - Z(p_0))^2}{s_3^2}\right) \tag{11}$$

where $Z(\mathbf{p})$ denotes the range component (z) of a pixel, s_3 defines the weight scale in depth coordinates and the kth sampled pixel corresponds to the i,jth component of $\mathbf{S}(\theta)$.

Next eigenvalue-eigenvector decomposition is applied for $\Sigma(\theta)$:

$$\Sigma(\theta) = \mathbf{V}(\theta)\Lambda(\theta)\mathbf{V}^T(\theta) \tag{12}$$

The eigenvector of the smallest eigenvalue is selected as the normal $\mathbf{n}(\theta)$. The smallest eigenvalue corresponds to the least significant direction, which is the normal of the best fitting plane at θ direction:

$$i^*(\theta) = \arg\min_{i=1..3}(|\Lambda_{i,i}(\theta)|) \tag{13}$$

$$\lambda^*(\theta) = |\Lambda_{i^*(\theta),i^*(\theta)}| \tag{14}$$

$$\mathbf{n}(\theta) = \mathbf{V}(\theta)^{i^*(\theta)} \tag{15}$$

At a given $\mathbf{p}_0$ point the normal $\mathbf{n}^*$ is selected which was estimated with the best fit:

$$\mathbf{n}^* = \mathbf{n}(\arg\min_{\theta}(\lambda^*(\theta))) \tag{16}$$

By using a-priori information about edges (position and orientation), the rotation process may be ignored and the appropriate $\mathbf{S}(\theta_{edge})$ sampling matrix may be used. As mentioned in Section 2 other methods exist for normal estimation, which are based on different error functions. Such methods can be easily integrated with the sampling technique given in this paper. PCA was selected to provide a baseline algorithm.

3.3 Edge detection

Edge detection in range images differs from how edges appear in intensity images. Edge types have been summarized in Section 2.

In our previous research [10] we have shown a method for detection and classification of edges in such discretized range images. The method was also based on skeleton extraction. Jump edges were detected by evaluating pixel local neighborhood for significant depth changes and taking quantization levels into account.

In case the range map was acquired using stereo disparity map, the quantization function is usually not linear, but may be already known or it can be identified from the image itself.

For jump edge detection $Z_{th}(z) = \sigma dZ(z)$ is selected where $dZ(z)$ is the quantization step at given z depth, $\sigma > 1$. Equation

$$\max_{||p-p_0||<r}(Z(p) - Z(p_0)) > Z_{th}(Z(p_0)) \tag{17}$$

is satisfied near jump edges on the foreground surface. To deal with false edges we not only look the adjacent pixels, but in case of missing data the adjacent side of the unknown region.

Crease edge detection was based on abrupt changes of skeleton orientation. Such orientation and distances between skeletons encode the surface normal as the orientation is a projection of the normal to the xy plane, distances carry information of the z component. Changes in the orientation means changes in the normal but not vice versa. In order to detect all create edges normal reconstruction is needed.

We also propose a new method for crease edge detection. For surface normal estimation a rotated bilateral sampling was used, resulting in a normal function of rotation $\mathbf{n}(\theta)$. We assume that along planar surfaces the variance of such function is low, but on crease edges the variance increases. An edge measure e is introduced as the mean square error of normals at different rotations:

$$e = \frac{1}{N}\sum_{\theta\in\Theta}\left(\cos^{-1}(\mathbf{n}(\theta)\cdot\bar{\mathbf{n}})\right)^2. \qquad (18)$$

4 RESULTS

In this section results of the proposed algorithm is presented. Both synthetic and real captured data are evaluated [1].

4.1 Synthetic data

Sample scenes were constructed and rendered with ground truth data: surface normals were available directly from the modeling software. Values are usually estimated for skeleton pixels only but all pixels are filled in the image according to the closest skeleton pixel (in image space). This leads to incorrect visualization in many cases near endpoints of skeletons.

A simple bilateral-type of sampling is used in Figure 4. It can be seen that jump edges are correctly kept, but crease edges are smoothed. This is normal as the depth function part of the bilateral weight function still produces high weights because of relatively small depth differences on both sides of the edge. The sampling matrix size was adapting to the local density of the skeletons.

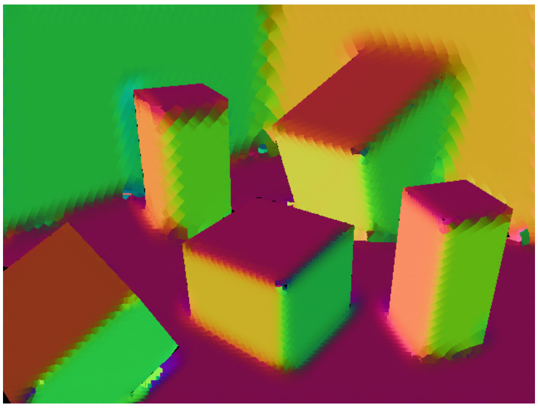


Figure 4: *Normal estimation using PCA and weighting based on distance in images space and depth. Boundaries are mostly kept intact, crease edges are blurred.*

Results using the proposed method is shown in Figure 5. Neither boundary nor crease edges were smoothed. The number of missing normals is very low. Some corners were smoothed.

Normal estimation error distribution is shown in Figure 6. Evaluating only skeleton pixels shows faster error fall-off as edges are not smoothed. The forward difference method shows slower fall-off, while the difference of the error distribution for all pixels or skeleton pixels do not show significant difference. For the proposed method the distribution for all pixels is similar to the compared method but still shows less uncertainty.

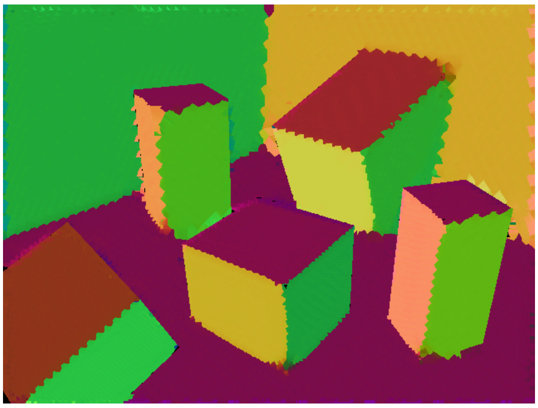


Figure 5: *Normal estimation using the proposed method. Neither boundaries nor crease edges are smoothed. The number of unestimated normals is minimal.*

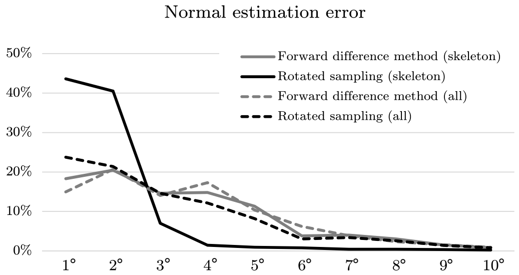


Figure 6: *Normal estimation error distribution. Forward difference method is colored gray, proposed sampling method is black. Errors for only skeletons pixels are shown as solid, for all pixels as dashed lines.*

Identified crease edges are shown in Figure 7. Crease edges are highlighted very well while object boundaries are not highlighted due to the bilateral behavior of the sampling matrix.

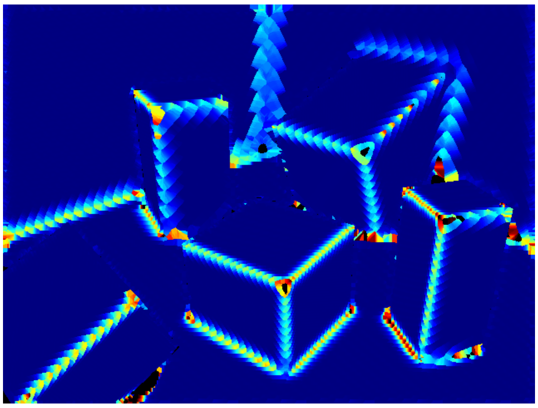


Figure 7: *Crease edge detection using edge measure given in equation (18)*

[1] http://dx.doi.org/10.6084/m9.figshare.1409432

4.2 Captured data

Data were acquired using the Microsoft Kinect sensor. In order to simulate even lower number of range layers the depth resolution was reduced manually. When observing a surface where the variance in depth direction is small, the resulting scene may also contain a low number of layers.

Figure 8 shows the reconstructed normal map of a captured scene using the forward differences method. The algorithm could not estimate normals around object boundaries due to limited skeleton information near borders and abrupt changes in skeleton directions cause incorrect estimates which are rejected. The range image contained only 28 depth layers. Due to noise normal estimation shows significant variance along planar surfaces.

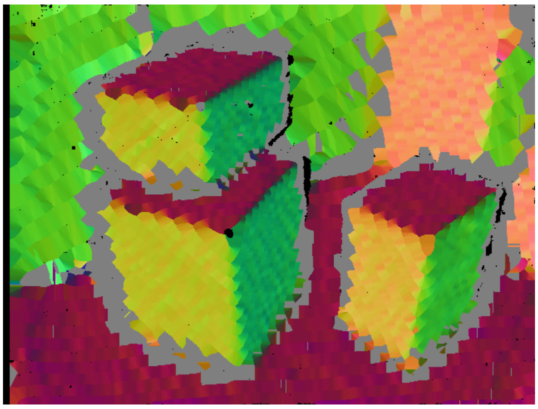


Figure 8: *Normal estimation based on forward difference method. Normal estimation failed around object borders. Crease edges appear smoothed.*

Results of the rotated bilateral sampling algorithm is shown in Figure 9. Edges are preserved and normals show less variance on planar surfaces. Such smoothing is due to the higher number of samples used in fitting compared to the one (or few) skeleton point on adjacent layer(s).

Estimated crease edges are shown in Figure 10. Due to noise and the very low number of range layers some false positive regions appear. Again this evaluation was run only on skeleton pixels but for visualization purposes data were interpolated using the nearest neighbor method. Hence the large curved positive regions.

5 CONCLUSIONS

In this paper we have shown a method for surface normal estimation and crease edge detection. The method is based on a set of sampling matrices that contain weights, and are constructed in advance as a function of size and orientation. Plane fitting is evaluated using adaptive size for the sampling matrix and also incorporating distance weights, similar to bilateral filtering.

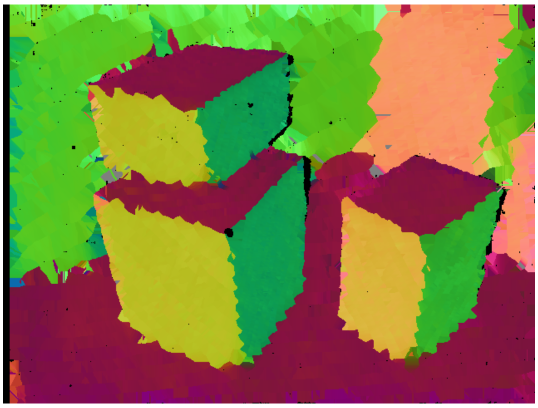


Figure 9: *Normal estimation based on the proposed rotated sampling and fitting. More edge points are preserved.*

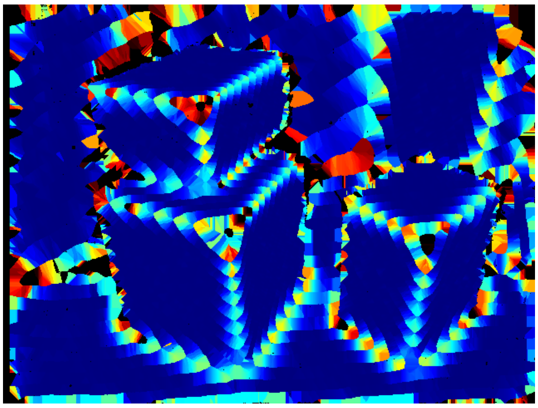


Figure 10: *Crease edge map estimated using the edge measure. Due to noise and the little number of layers false positive regions appear.*

This step produces a surface normal and a fitting error measure for each orientation. Simple statistics are used to select the best fitting plane and to identify a crease edge score. The method can be easily extended to incorporate other fitting methods: such as simple least squares, or apply two-step methods such as RANSAC.

We have presented examples of the output of the algorithm for both simulation and real data. Results show significant improvement compared to bilateral filtering as crease edges are less prone to blurring. We also compared a previously implemented method (forward differences between layers) that was used for normal estimation in heavily quantized range images. Although the proposed algorithm runs slower than the forward difference method, the results are more accurate: crease edges are less blurred and normal variation is lower on planar surfaces and data are estimated for more pixels. A method for crease edge detection was also presented based on the output of the normal estimation algorithm. Future research involves smoothing heavily quantized range images. Such results can successfully be used for further range image processing: segmentation, mapping, localization, object detection, recognition etc.

6 ACKNOWLEDGMENTS

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) organized by VIKING Zrt. Balatonfured. This work was partially supported by the Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund (grant no.: KMR 12-1-2012-0441).

7 REFERENCES

[1] H. Badino, D. Huber, Y. Park, and T. Kanade. Fast and accurate computation of surface normals from range images. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3084–3091, May 2011.

[2] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. A data structure for the 3d hough transform for plane detection. In *Proceedings of the 7th IFAC symposium on intelligent autonomous vehicles (IAV 2010), Lecce, Italy*, 2010.

[3] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nuchter. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2), 2011.

[4] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.*, 26(3), July 2007.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[6] S. Holzer, R.B. Rusu, M. Dixon, S. Gedikli, and N. Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689, Oct 2012.

[7] L. Iocchi, K. Konolige, and M. Bajracharya. Visually realistic mapping of a planar environment with stereo. In *Experimental Robotics VII*, ISER '00, pages 521–532, London, UK, UK, 2001. Springer-Verlag.

[8] A. Jose and C.S. Seelamantula. Bilateral edge detectors. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1449–1453, May 2013.

[9] K. Klasing, D. Althoff, D. Wollherr, and M. Buss. Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3206–3211, May 2009.

[10] V. Kovacs and G. Tevesz. Edge detection in discretized range images. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*, pages 203–208, Nov 2014.

[11] V. Kovacs and G. Tevesz. Plane segmentation in discretized range images. In *Workshops on Electrical and Computer Engineering Subfields, Proceedings of*, pages 184–189, Aug 2014.

[12] J. E. Kyprianidis and H. Kang. Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum*, 30(2):593–602, 2011.

[13] Anh Vu Le, Seung-Won Jung, and Chee Sun Won. Directional joint bilateral filter for depth images. *Sensors*, 14(7):11362–11378, 2014.

[14] Kai-Han Lo, Y.-C.F. Wang, and Kai-Lung Hua. Joint trilateral filtering for depth map super-resolution. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6, Nov 2013.

[15] G. Németh and K. Palágyi. 2d parallel thinning algorithms based on isthmus-preservation. In *ISPA 2011: 7th International Symposium on Image and Signal Processing and Analysis: Dubrovnik, Croatia, 4 - 6 September 2011. IEEE*, pages 585–590, 2011.

[16] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke. Efficient multi-resolution plane segmentation of 3d point clouds. In Sabina Jeschke, Honghai Liu, and Daniel Schilberg, editors, *Intelligent Robotics and Applications*, volume 7102 of *Lecture Notes in Computer Science*, pages 145–156. Springer Berlin Heidelberg, 2011.

[17] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[18] Qingxiong Yang, N. Ahuja, Ruigang Yang, Kar-Han Tan, J. Davis, B. Culbertson, J. Apostolopoulos, and Gang Wang. Fusion of median and bilateral filtering for range image upsampling. *Image Processing, IEEE Transactions on*, 22(12):4841–4852, Dec 2013.

[19] Mincheol Yoon, Yunjin Lee, Seungyong Lee, Ioannis Ivrissimtzis, and Hans-Peter Seidel. Surface and normal ensembles for surface reconstruction. *Computer-Aided Design*, 39(5):408 – 420, 2007.

Surfaces for Point Clouds using Non-Uniform Grids on the GPU

Daniel Schiffner
Goethe Universität
Robert-Mayer-Strasse 10
60054, Frankfurt, Germany
schiffner@gdv.cs.uni-frankfurt.de

Claudia Stockhausen
Goethe Universität
Robert-Mayer-Strasse 10
60054, Frankfurt, Germany
stockhausen@gdv.cs.uni-frankfurt.de

Marcel Ritter
AHM, Universität Innsbruck
Technikerstrasse 21a
6020, Innsbruck, Austria
marcel.ritter@uibk.ac.at

ABSTRACT

Clustering data is a standard tool to reduce large data sets, such as scans from a LiDAR, enabling real-time rendering. Starting from a uniform grid, we redistribute points from and to neighboring cells. This redistribution is based on the properties of the uniform grid and thus the grid becomes implicitly curvilinear, which produces better matching representatives. Combining these with a polygonal surface reconstruction enables us to create interactive renderings of dense surface scans. Opposed to existing methods, our approach is running solely on the GPU and is able to use arbitrary data fields to influence the curvilinear grid. The surfaces are also generated on the GPU to avoid unnecessary data storage.

For evaluation, different data sets stemming from engineering and scanning applications were used and have been compared against typical CPU based reconstruction methods in terms of performance and quality. The proposed method turned out to reach interactivity for large sized point clouds, while being able to adapt to the point clouds geometry, especially when using non-uniform sampled data.

Keywords
Surface reconstruction, point clouds, clustering, curvilinear grids

1 INTRODUCTION

With the increasing use of laser light detection and ranging (LiDAR), applications easily generate several billions of points measurements [PMOK14] [OGW$^+$13]. If semi-automated algorithms are to be applied, interactive rendering of such large data sets becomes important. However, such large amounts of geometry data do not fit into the graphic hardware's memory as they easily reach hundreds of giga-bytes. While out-of-core mechanisms are used, the generation of the needed information cannot be solely computed on the graphics card.

Our approach addresses one part of the overall problem by generating a representation that allows interactive rendering. We want to avoid as many pre-processing steps as possible, and thus use a clustering algorithm as our base method for data reduction. We leverage the final grid definition to modify the grid using per-cell information, resulting in a curvilinear representation. Following these steps, we are then able to use these curvilinear cells to produce better fitting surface-representations based on cell-only information. In this work, we enhance our previous definition and contribute the following:

- Refined and more precise definition of the curvilinear grid.
- Simple and robust surface normal estimation
- Different, dynamic reconstruction methods, ranging from cells to surfaces

After providing some background information, we gather related and previous work in section 2. The approach is described in-depth in section 3 followed by an evaluation and results in section 4. Here, the main results regarding to timing and visualization are presented and discussed. The article concludes in section 5, and closes with thoughts on future work in section 6.

2 RELATED AND PREVIOUS WORK

The application of vertex clustering recently has grown in interest due to its fast processing capabilities. Linear methods, such as grid based clustering methods, are especially well suited for large data sets that may contain several million or even billion data points. By reducing the input set, such as presented by DeCoro [DT07] or Willmot [Wil11], the rendering of large data is possible again with a little overhead at the initial clustering phase. In the latter case, individual attributes of an input data set are stored separately to increase detail after reduction.

Promising results have also been achieved by Peng and Cao [PC12], as they are able to provide frame-to-frame coherence when applying their reduction method. Their approach is based on an edge collapse algorithm, which was presented by Garland and Heckbert [GH98]. They apply the computation of the quadric error metric in parallel and then decide where to reduce and restructure the output triangles.

The selection of individual level of details is also a crucial part and often includes offline processing methods. In [SK12], we used a parallel approach to dynamically update the current representation while retaining interactivity. This is done by first computing a raw estimate of the object that is being refined during processing. In our previous work [SRB14], we have used the *cluster move* paradigm to enhance clustering of unstructured point clouds. In Limper et al. [LJBA13], the so-called POP Buffer has been introduced. The authors make use of a simple clustering followed up by a sorting on the CPU. This allows fast LOD exchange, as solely VBOs are created that can also be instanced. None of the presented approaches makes use of the processing capabilities provided by modern GPUs.

A comparison of two clustering algorithms has been presented by Pauly [PGK02]. In this case, a hierarchical and an incremental clustering method are applied to reduce and refine point-set-surfaces[ABCO$^+$01]. Both approaches showed good results regarding reduction quality and run-time performance. Clustering, especially in the context of SPH data sets, has been utilized by [FAW10] with a perspective grid. They include a hierarchy (octree) in the data organization and apply texture based volume rendering in front to back order of the perspective grid for drawing.

[PGK02] use a covariance technique in the point neighborhood to compute a reconstructed 'surface normal' and to measure a distance from a cluster point to the original surface. A similar method based on the same dyadic product, called the point distribution tensor, was introduced in our previous work [RB12]. However, the product also includes distance weighting functions and the analysis based on the tensor's Eigenvalues is different. Three scalar fields are derived from the second order tensor called linearity, planarity, and sphericity. These describe the geometric point neighborhood and are normalized between 0 and 1. If points are distributed on a straight line, linearity is high, and if points are distributed on a plane planarity is high, respectively. We pre-computed the planarity for some of the data sets used in the benchmarks and include it in the clustering process, such that variations in planarity are preserved and homogeneous planar regions are clustered.

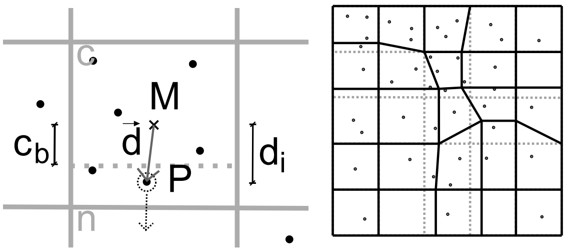


Figure 1: **Left:** Detailed view of the marked cell in Figure 2. This graphic illustrates how a point is "moved" from one cell to its neighbor below. A point P of the cell c is assigned to a different cell if the largest component d_i of the direction vector $\vec{d}$ from the cell center M to P is larger than a certain cell bound c_b which depends on parameters of cell c and the neighboring cell n. **Right:** Curvilinear grid after moving the points. Note that the curvilinear grid is not computed explicitly. It is indirectly defined by the points being assigned to the cells.

Besides the area of workstations the problem has also been addressed for mobile devices. Rodríguez et al. [RGM$^+$12] presented a method for interaction with giga-sample-sized point clouds on mobile devices. The solution is based on a server-client framework, with a previous pre-processing step. In the pre-processing step, the data is partitioned as a kd-tree and reduced to a target point size. The points are reduced by merging points with their closest neighbor with compatible normals. For a further overview of 3D graphics on mobile devices we refer to Koskela et al. [KVA15]

3 OUR APPROACH

Our approach makes use of a simple vertex clustering that can be computed and generated on the graphics card. This initial clustering is used in a second iteration to resize individual cells. The available information hereby ranges from simple density distributions up to more complex scenarios, where planarity is being computed.

We enhance our previous definition [SRB14] by providing a more complex, but more powerful function to compute the cell boundaries. We also pass surface information gathered during the clustering-stages to later stages of the pipeline to allow dynamic reconstruction of surfaces. The resulting surfaces are only created within the graphics card, but can also be retrieved using transform feedback mechanics.

It is noteworthy, that the source data is not altered during the computation of the cell boundaries. The processed vertex is only passed to a different cell within the second iteration.

The presented approach works in multiple steps: *cluster*, *move*, *reduce*, and *reconstruct*. *Cluster* and *move* as well as *reduce* and *reconstruct* are pairwise related. For this reason, they will be introduced together in the following sections.

3.1 Cluster and Move

The *cluster* operation applies a classical vertex clustering, but we also accumulate information required for the *move* operation. The incoming points are mapped to a uniform grid. The grid might be warped via an affine transformation as shown in figure 2. The point position is converted to an cell index that is used in further computations.

The *move* operation then identifies whether a point needs to be placed in a neighboring cell. Based on the accumulated information, e.g the number of points inside a cell and the position, curvilinear cell bounds are derived. If the current point is located outside the curvilinear cell bound, it is emitted as being part of this neighbor.

The given definition of the cell boundaries is curvilinear, as a formula is given that modifies the actual borders. This is also illustrated in figure 1. In the following, we provide the formula given by [SRB14]:

$$\begin{aligned}
\vec{d} &= P - M && (1)\\
\Delta_i &= \max_{j=1..3}\{|d_j|\} && (2)\\
\mathrm{w}(c,n) &= \min\left(lb, \left(\frac{dens(c)}{dens(c)+dens(n)}\right)^{\gamma}\right) && (3)\\
c_b &= w(c,n) && (4)\\
\Delta_i &> c_b \begin{cases} true, & move\ P \text{ to } n \\ false, & \text{skip} \end{cases}, && (5)
\end{aligned}$$

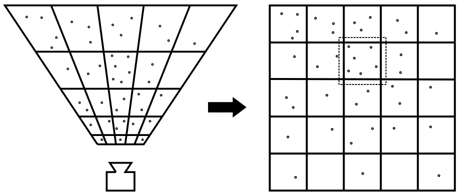


Figure 2: Points are transformed into a local coordinate system of the camera view frustum. Initial cells are defined by a uniform grid. The clustering algorithm operates in this coordinate system. The grid's geometry preserves more detail close to the camera and reduces detail far way.

with M the center point of the current cell c, P a point in c, i the index of the maximal component of vector $\vec{d}$, n the neighbor cell, lb a lower bound of the cell size of c, c_b the cell boundary in direction of the component i, and γ a non-linear scaling factor.

For a point P, its direction vector $\vec{d}$ from the cell's center is computed. The maximal absolute component of this vector is chosen, see Equation 1 and 2. Then, a weight dependent on the current cell c and its neighbor n, a lower bound and a non-linear scaling factor is computed, defining a new theoretical cell bound c_b, see Equation 3 and 4. If the maximal component is larger than this new cell bound, the point is assigned to the neighbor cell, see Equation 5.

The $dens(x)$ function operates on information available inside a cell x and returns a scalar value. For data sets having only point coordinates available, we use the number of points inside a cell as density. Any data available per point can be included in the density function. We also experimented using a pre-computed planarity field [RB12]. This field describes a local geometric property of the point cloud, influencing the *move* operation.

The computational complexities of the *cluster* and *move* operations scale with the size of the input data $\mathcal{O}(N)$. Each cell, identified by the index, is processed and data is accumulated per cell.

To apply this function to 3d objects, we enhance the definition by replacing them by

$$\begin{aligned}
\Delta &= P - M && (6)\\
f(c,n) &= 2\frac{dens(c)}{dens(c)+dens(n)} && (7)\\
w(c,n) &= clamp\left((f(c,n))^{\gamma}, lb, 1-lb\right) && (8)\\
idx_c &\mathrel{+}= \sum_{i=0}^{3}(w(c,n_i) < \Delta_i)\cdot idx_{n_i} && (9)
\end{aligned}$$

The function expands the current relative weights to the range of $[0,1]$ and limits it to the resulting range of the lower bounds. The delta is used to relate the new boundaries with the current location of the inspected point. The resulting index of the current point (idx_c) is computed by the current index plus the possible offsets (idx_{n_i}). A Boolean value of false is "0", while a Boolean value of true is "1", like in C.

In either case, a piecewise-linear shift for each cell boundary is created. However, due to the fact, that each cell has its own relative weights, these boundaries are still curvilinear.

3.2 Reduce and Compute Normals

The *reduce* operation emits an representative for each cell that has been previously computed. Thus, the output is a reduced set of points. Any accumulated data can be emitted and visualized as well. As the single cells are iterated, the computational complexity is bound linearly with the number of cells $\mathcal{O}(C)$. After this reduction, the visualization of the reduced data set can be done using classical splatting techniques.

To reconstruct surfaces, we derive a surface normal along with the boundaries of the current cell. We are able to approximate the surface normals by using two different approaches. The first method leverages the point-distribution tensor, whereas the second uses the cell neighborhood. In the former case, the *reconstruction* operation computes the minor eigenvector and uses it as the surface normal.

The surface normal approximation based on the cells uses the density information gathered in the *move* operation. We hereby assume, that the surface normal will be oriented from a cell towards empty cells. Thus we iterate the neighboring cells, and add up directions, where the neighboring cells are empty. This yields a surface normal, which points outwards from the current cell. If all neighboring cells are empty, no normal can be derived (which is correct).

Independent of the used approach, the estimated normal vector has no preferred direction and has to be oriented. We therefore use the positive-z axis in eye space, favoring normal vectors that are facing towards the observer.

3.3 Dynamic Reconstruction

Based on the averaged cell position, the bounding box and the optional estimated normal vector, we create geometric representatives for each cell. We use three different geometric objects: boxes, oriented splats, and cell-filling-quads. One such geometric representative is generated per cell. The boxes can the created from the position and the bounding box, whereas the oriented splats and the cell-filling-quads require the estimated normal vector for correct vertex orientation.

The splats are given the radius based on the distance to the neighboring cells, which can easily be derived to the fixed relationship of the cells. Along with the estimated surface normal, a perspective correct splatting can be achieved.

In the case of the oriented surfaces, we use the surface normal, to map 4 vertices in the bounding box. The resulting positions are thus limited by the bounding box of the current cell.

All methods share, that the created geometry is created on the GPU, and no transfer or storage of this data is required. By leveraging geometry shaders, the generation of new primitives can be achieved efficiently, as the maximal number of primitives is limited by the grid resolution used during the vertex clustering stage. Thus, we can derive a maximal number of vertices and ensure interactive visualizations by limiting the grid size.

4 RESULTS

To create test results, we have implemented our approach with openGL using compute shader capabilities that are available since version 4.3. We did not use an openCL approach, as the data will be rendered directly after the processing. This way, we have direct control on the results of the cluster algorithm when altering the individual parameters. In the core specification, no floating point atomic operations are specified but can be added by using an extension from nVidia. When using other vendors, one could emulate this feature, by converting the float value to an integer. For further details, the reader may be referred to [CCSG12].

As our approach consists of a *cluster* and a *move* step, we can simply omit the latter to allow an evaluation of the overhead generated. Thus, this algorithm applies a basic clustering to the input data set. A top-down octree has been implemented using the CPU. Obviously, the octree will not be able to compete in terms of computation speed, but the reduced cells are used for a visual comparison. We opted to use an octree because the clustering methods presented in the background section either require heavy precomputations ([PGK02], [FAW10]) or use a hierarchical clustering ([DT07],[Wil11]). In the latter case, especially regarding the work of [Wil11], we do not have several attributes for our data, thus we cannot make use of the advantages of this algorithm.

While processing large data sets, one must take special considerations into account. One being the limitations of the used graphics card. The compute shader capabilities have several, graphics card dependent factors, such as maximal work group size or maximal buffer size. The latter is especially important for large data sets, as a streaming of individual data is necessary. The approach is able to compute partial solutions, as the grid can be constructed in a streaming fashion. In the case of the largest data set (refer to 1), the computation times partially reached a Windows specific timeout (TDR), where the driver is shutdown and restarted. We use a swap of the back buffer to circumvent such an timeout after each step. While not being optimal, as the graphics is busy swapping a buffer, it allows to keep the driver alive. Similar problems are known when using expensive shaders of any type, CUDA and OpenCL applications. Additionally, an out-of-core mechanism is required, if the used data exceeds the maximal buffer size of the GPU. However, this is not taken into account yet.

4.1 Time Measurement Results

Based on our application, several benchmarks have been conducted. They vary in terms of input size, grid size and used graphics card. In general, a test has been repeated 10 times and the median time values are given.

Timings are reported in milliseconds. Each test was run with varying input parameters, i.e. the object and the grid size. These benchmarks were executed on 3 different PC's, running on Windows 7 and Linux. The results are listed in table 1. The first system (1) uses an i5-670 and a nVidia GeForce 680 GTX. The second (2) uses an i5-333 with a GTX 780 Ti. The last machine (3) consists of an i5-3450 and a nVidia GeForce GTX 460 with 1GB RAM. All systems operate on a MS-Windows platform.

The results in table 1 are split into two sections, the grid based operations and the visualization. The latter uses a standard view, to be comparable among the different tests. The number of reduced elements is given in the first row of each data set. Note, that the test system 3 is running at its maximum capabilities, due to the available memory, and could not process the last data set trivially. For this reason, we have excluded it from the benchmark, as an out-of-core strategy needs to be used.

The individual timings indicate an overhead due to the additional processing step of our approach. We have an increase of approximately 100% if the *move* operation is used. Note, however, that our compute shader has not been optimized and leaves room for further improvements. Currently, the *move* operation does a complete reclustering of the source data instead of using the results of the first stage.

A visualization of the presented timings using a different grid size can be seen in Figure 3. Interestingly, the computation times reduce, as the grid increases in size. This is mainly due to the fact, that an atomic counter is required, once an element is emitted into a cell. The smaller the overall cell count, the more atomic writes into an individual cell are required. This results in more sequential writes in this case.

As one can easily see in Figure 3, the GeForce 460 GTX is not able to compete with the newer generations. This may be due to the limited memory as well as being the first generation supporting compute shader capabilities.

To further emphasize the influence of each individual processing step, i.e. *cluster*, *move* and *reduce*, we created a visualization of these steps in figure 4. The *move* operation uses approximately the same time as the *clustering*. The influence of the neighborhood size, i.e. zero, one or three, is negligible. The reduction in this case uses the point-tensor reconstruction, which may be opted out for an rougher and faster approximation.

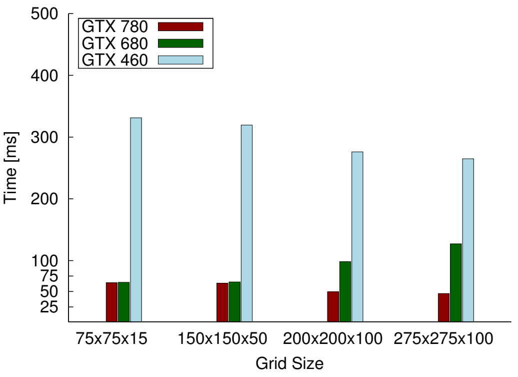


Figure 3: The influence of the grid size on the overall performance of our algorithm. The GeForce 780 GTX outperforms the other graphics cards. The 460 GTX is not able to compete with any of the newer versions. We used the GasTank data set for computation.

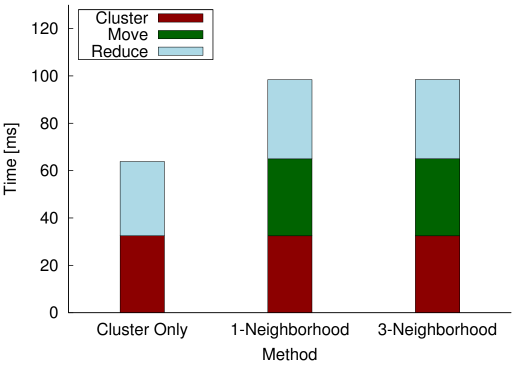


Figure 4: Timing values for each processing step of our algorithm. The reported values are the median of all runs. For this values, the medium sized object with a grid size of 200x200x100 has been used. Measurements were taken on the Test System 1.

4.2 Visual Results

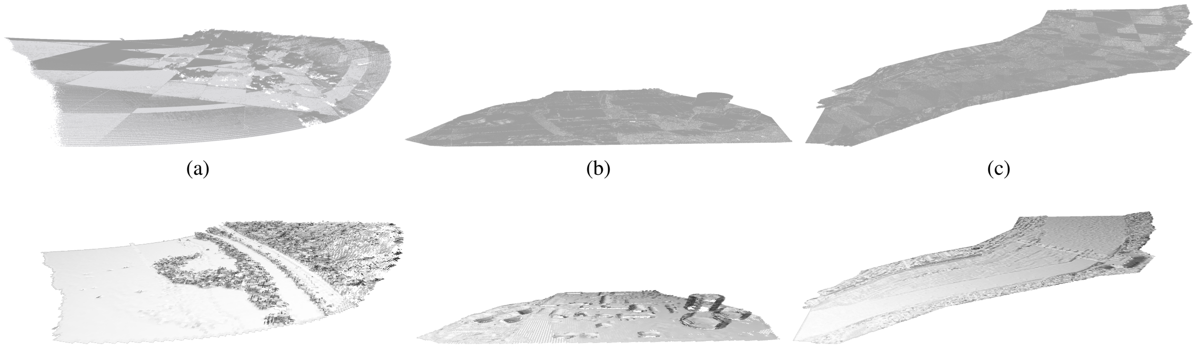
The visualization technique draws either oriented splats using elliptical weighted average (ewa) splatting [ZPvBG02], a cell representation based on boxes or a surface approximation, as described in section 3.3. In figure 5 some results generated with our surface reconstruction method are shown. We used the prior mentioned data sets to apply a clustering. The number of generated primitives is significantly lower than the input count, thus achieving (in general) much higher frame rates (refer to table 1).

The different methods of visualizations are generated using a geometry shader using information stemming from the bounding boxes and the normal estimation. In case of the boxes representation, the normal estimation step can be skipped. The other methods require a surface normal for orientation. An overview of the results using the Small River data set can be seen in figure 6.

As mentioned before, we use an octree implementation to show differences in restruction quality of our approach. We selected an octree level, which approximately generates the same number of primitives as our approach, as seen in figure 7. Our algorithm is able to create similar results, but is much more flexible and can be solely computed on the GPU, which is not trivial to achieve with the octree approach.

Model	Sys	Our	Cluster	CPU	Original	Splat	Boxes	Surfaces
SmallRiver					# 2.075.993	# 75.173	# 902.076	# 150.346
	1	62.8	46.7	873.0	7.6	2.3	10.2	3.8
	2	63.1	46.6	180.0	4.1	2.8	11.0	5.0
	3	68.3	52.0	790.0	7.7	46.6	77.9	48.1
GasTanks					# 11.133.482	# 67.993	# 815.916	# 135.986
	1	65.7	49.3	4753.2	30.3	2.2	9.5	3.8
	2	64.3	43.5	940.0	18.3	3.1	11.0	5.0
	3	318.7	267.5	4110.0	37.0	46.3	79.7	48.4
RiverDam					# 26.212.555	# 79.099	# 949.188	# 158.198
	1	123.9	89.2	11006.6	71.8	2.2	9.3	3.8
	2	97.9	65.1	2234.9	45.6	2.8	13.7	5.1

Table 1: Benchmark results of our GPU algorithm, a basic cluster approach, an CPU and an octree implementation. All shown tests have been performed with a grid size of 150x150x50. Timings are reported in ms, the numbers in the first row of each data set denote the number of elements used for visualization. Note that the test system 3 was not able to perform the clustering due to hardware limitations, which may be the reason for the bad timings in the visualization, despite the low number of vertices used.


(a) (b) (c)

(d) SmallRiver (e) GasTank (f) RiverDam

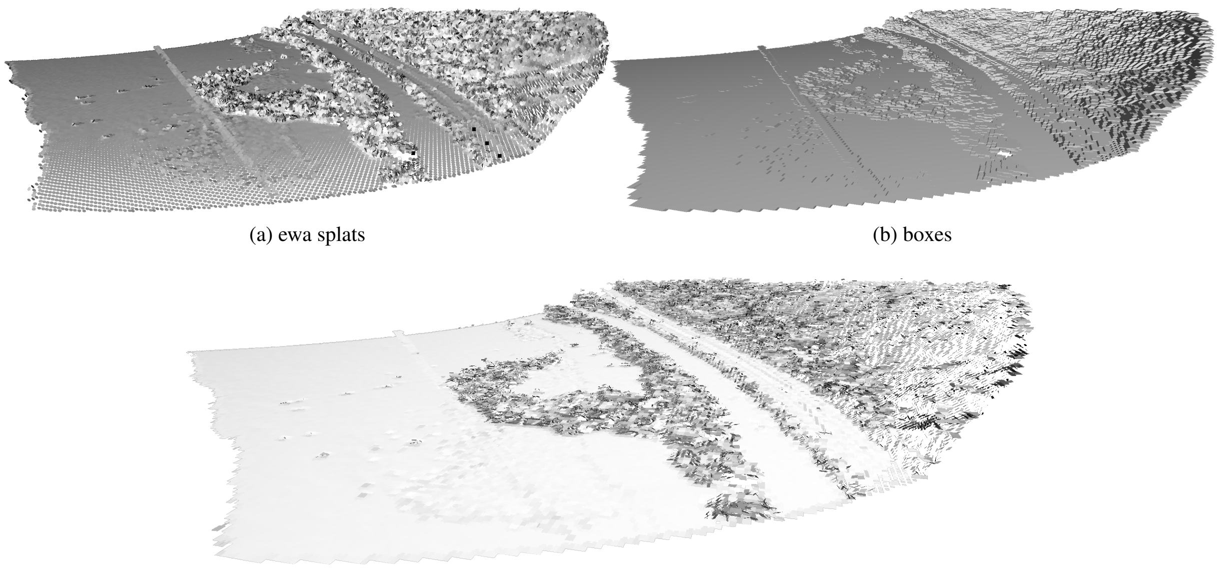
Figure 5: **Top row:** The used three point cloud data sets used for testing. Different sizes and different geometrical distributions are benchmarked. The points in the LiDAR data sets are mainly distributed on surfaces with small volumetric regions in vegetation and water. The point density varies relatively little over the whole data set.
Bottom row: Visual results of the surface reconstruction using clustering for the different data sets illuminated using a headlight. The grid size is set to $150 \times 150 \times 50$.

Both methods used for normal estimation provide stable results. The raw approximation based on the reduced cell set has a significant lower processing time (about 40%), but the tensor method provides much smoother and higher quality normals. A comparison of the achieved results can be seen in figure 8. The holes in both images method appear, as the cell-filling quads are aligned on the surface normal.

We lastly tested the influence of the *move* operator and the number of neighbors used, show in figure 9. As seen in the timing measurements, the number of neighbors does not significantly changes the processing time of the *move* operator. The quality of the surfaces increases with the larger number of neighbors used in the curvilinear grid computation. In the case of 3-neighbors, the discontinuity in the scan can better be represented due to the better matching cells, than in the other two cases. While the gaps between the individual quads disappear, there are still misplaced patches. These arise do to missing neighborhood information in these regions. Thus, the normal computation fails to derive an unambiguous direction, yielding these misplaced patches. By increasing the neighborhood, these cases can be avoided but with the cost of more expensive computations.

5 CONCLUSION

We have presented a new approach to reconstruct surfaces by leveraging a non-linear clustering to arbitrary objects. We are able to use multiple information from the current geometry and are not limited any preprocessing. The applied reduction is made selectively, due to a restructuring of individual cells. Currently, our data sets are point based and do not incorporate connectivity information. However, an extension to triangles or polygons can easily be achieved, as shown by other researchers ([PC12] [Wil11]).

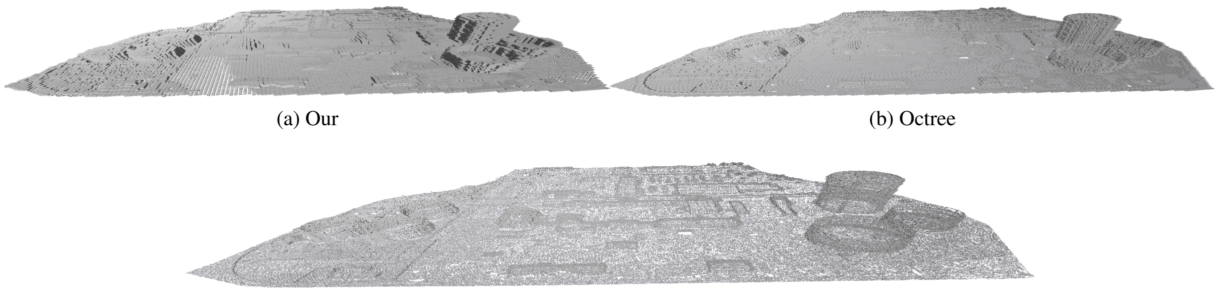


(a) ewa splats

(b) boxes

(c) surface

Figure 6: The small river data set geometrically reconstructed on the GPU. A geometric object is created per cell based on its information, i.e. center point, density, cell-size, and normal vector. The top row illustrated ewa splats and boxes. The bottom figure shows result of our surface reconstruction method. The model is illuminated using a headlight.



(a) Our

(b) Octree

(c) Octree high detail

Figure 7: Comparison of our approach and an octree reconstruction. The achieved quality is very similar, while our approach is created solely on the GPU and does not require any precomputations. The high detail representation is included to show the desired result. Note that the high detail representation fails to create a closed surface, as the leaf-level of the octree does not fill the resulting gaps.

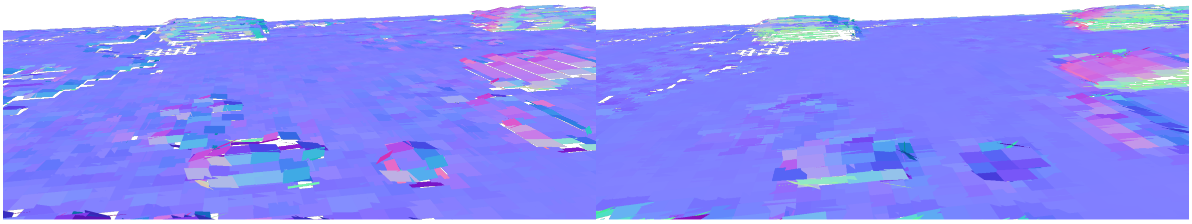
The computation times of the *cluster move* operations are interactive for medium sized point clouds and has a good performance with large data sets. Our implementation has not been optimized and leaves room for further enhancements.

We have shown the differences between classical clustering and our curvilinear implementation. Due to the dynamic cells, details in an object are better preserved. This increases the quality during a rendering and represents the topology of the basic object more accurately, while still reducing the input data set.

The reconstruction of a surface based on the geometric properties of an individual cell allows different visualizations without the need of re-computation. We use accumulated information of the cluster cells to create simple per-cell geometric elements to approximate a surface. We demonstrated three different elements on a LiDAR data set, allowing to reconstruct a polygonal digital surface model in real-time.

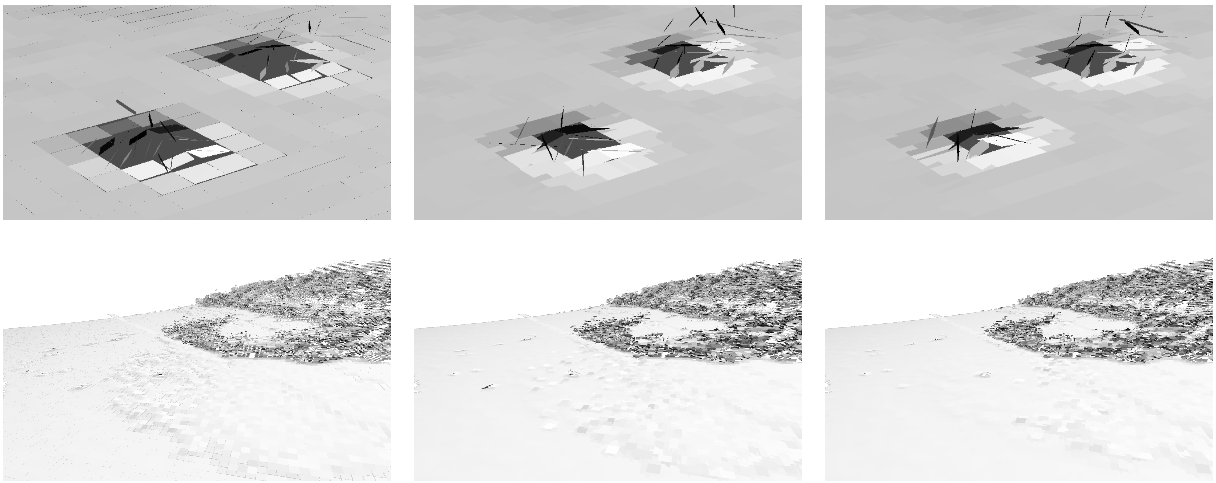
6 FUTURE WORK

The high performance of the compute shader drives us to further investigate streaming of big data. This includes a fast discard of unnecessary data, as well as selective reloading of individual fragments of a rendered object. Especially, the efficiency of the *move* allows repetitive execution (more iterations) or more complex grid modifications. The current restrictions to direct neighbors can be removed with the cost of additional lookups during the *move* operation. We assume that this will further improve the quality of the clustering.



(a) Normals using empty cells (b) Normals using point tensors

Figure 8: Comparison of our normal computations based on the reduced cells. The surface normals based on the tensor are more stable and produce more smooth approximations, but has significantly higher computational complexity. As one can see, the planar regions show less jitter in the tensor case. Both images were created using the cell-filling quads for visualization.



(d) Cluster only (e) Single neighbor (f) Three neighbors

Figure 9: Comparison of the different neighborhood computations. On the left, the *move* operation has been skipped, i.e. cluster only), where as the middle image shows the single neighbor result. In the right picture, 3 neighbors were taken into account, resulting in less gaps in the visualization. The outliers arise due to the surface normal estimation, which is performed in an online manner. The lower row shows an overview of the data set.

As the cell-filling quads are an interesting starting point, to polygonalize a surface, we think, that curved surfaces, such as Bézier or NURBS patches, do better match the underlying geometry. However, further testing needs to be done, how these patches can be utilized without any larger pre-processing of the data set. Furthermore, these patches could be controlled in terms of level of detail by a tessellation-shader, further enhancing the dynamic reconstruction.

An interesting topic is the dynamic construction of reusable information by defining a maximal footprint of GPU memory to use. In this case, LoD algorithms need to be applied, to ensure correct selection and eviction of primitives for display.

We will improve the quality of the tensor computation, especially by investigating better points of reference than the center point of a cluster cell. We intent to represent more information gathered in the tensor in the geometrical reconstruction. We will enable the reconstruction of linear structures besides planar ones. We want to improve the cell-filling-quads generation to better represent partially smooth closed surfaces.

Acknowledgements. This work was supported by the Austrian Ministry of Science BMWF as part of the Konjunkturpaket II of the Focal Point Scientific Computing at the University of Innsbruck and as part of the UniInfrastrukturprogramm of the Research Platform Scientific Computing at the University of Innsbruck and funded by the Austrian Science Fund (FWF) DK+ project Computational Interdisciplinary Modeling, W1227-N16. We like to thank Frank Steinbacher [ahm15] to provide the LiDAR data sets.

7 REFERENCES

[ABCO$^+$01] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Cláudio T. Silva. Point Set Surfaces. In Thomas Ertl, Kenneth I. Joy, and Amitabh Varshney, editors, *IEEE Visualization*. IEEE Computer Society, 2001.

[ahm15] 2015 (accessed March 9, 2015). http://ahm.co.at.

[CCSG12] Cyril Crassin and Simon Green. Octree-Based Sparse Voxelization Using the GPU Hardware Rasterizer. In Patrick Cozzi and Christophe Riccio, editors, *OpenGL Insights*, pages 303–319. CRC Press, July 2012. http://www.openglinsights.com/.

[DT07] Christopher DeCoro and Natalya Tatarchuk. Real-time Mesh Simplification Using the GPU. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, pages 161–166, New York, NY, USA, 2007. ACM.

[FAW10] Roland Fraedrich, Stefan Auer, and Rüdiger Westermann. Efficient high-quality volume rendering of sph data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1533–1540, 2010.

[GH98] Michael Garland and Paul S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization*, pages 263–269, 1998.

[KVA15] Timo Koskela and Jarkko Vatjus-Anttila. Optimization techniques for 3d graphics deployment on mobile devices. *3D Research*, 6(1):1–27, 2015.

[LJBA13] M. Limper, Y. Jung, J. Behr, and M. Alexa. The pop buffer: Rapid progressive clustering by geometry quantization. *Computer Graphics Forum*, 32(7):197–206, 2013.

[OGW$^+$13] Johannes Otepka, Sajid Ghuffar, Christoph Waldhauser, Ronald Hochreiter, and Norbert Pfeifer. Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. *ISPRS International Journal of Geo-Information*, 2(4):1038–1065, 2013.

[PC12] Chao Peng and Yong Cao. A GPU-based Approach for Massive Model Rendering with Frame-to-Frame Coherence. *Comp. Graph. Forum*, 31(2pt2):393–402, May 2012.

[PGK02] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient Simplification of Point-sampled Surfaces. In *Proceedings of the Conference on Visualization '02*, VIS '02, pages 163–170, Washington, DC, USA, 2002. IEEE Computer Society.

[PMOK14] N. Pfeifer, G. Mandlburger, J. Otepka, and W. Karel. OPALS - A framework for Airborne Laser Scanning data analysis. *Computers, Environment and Urban Systems*, 45(0):125 – 136, 2014.

[RB12] Marcel Ritter and Werner Benger. Reconstructing Power Cables From LIDAR Data Using Eigenvector Streamlines of the Point Distribution Tensor Field. *Journal of WSCG*, 20(3):223–230, 2012.

[RGM$^+$12] Marcos Balsa Rodriguez, Enrico Gobbetti, Fabio Marton, Ruggero Pintus, Giovanni Pintore, and Alex Tinti. Interactive exploration of gigantic point clouds on mobile devices. In *VAST*, pages 57–64, 2012.

[SK12] Daniel Schiffner and Detlef Krömker. Parallel treecut-manipulation for interactive level of detail selection. In *20th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, volume 20, 2012.

[SRB14] Daniel Schiffner, Marcel Ritter, and Werner Benger. Using curvilinear grids to redistribute cluster cells for large point clouds. *Proceedings of SIGRAD 2014*, pages 9–16, 2014.

[Wil11] Andrew Willmott. Rapid Simplification of Multi-Attribute Meshes. In *High-Performance Graphics 2011*, August 2011.

[ZPvBG02] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus H. Gross. Ewa splatting. *IEEE Trans. Vis. Comput. Graph.*, 8(3):223–238, 2002.

VideoMR: A *Map and Reduce* Framework for Real-time Video Processing

Benjamin-Heinz Meier
Hasso-Plattner-Institute
University of Potsdam,
Germany
benjamin-heinz.meier
@student.hpi.de

Matthias Trapp
Hasso-Plattner-Institute
University of Potsdam,
Germany
matthias.trapp@hpi.de

Jürgen Döllner
Hasso-Plattner-Institute
University of Potsdam,
Germany
juergen.doellner@hpi.de

ABSTRACT

This paper presents VideoMR: a novel map and reduce framework for real-time video processing on graphic processing units (GPUs). Using the advantages of implicit parallelism and bounded memory allocation, our approach enables developers to focus on implementing video operations without taking care of GPU memory handling or the details of code parallelization. Therefore, a new concept for map and reduce is introduced, redefining both operations to fit to the specific requirements of video processing. A prototypic implementation using OpenGL facilitates various operating platforms, including mobile development, and will be widely interoperable with other state-of-the-art video processing frameworks.

Keywords

map and reduce, video processing, real-time, bounded memory

1 INTRODUCTION

Modern video processing has been shifted from post-processing to real-time systems, applications, and techniques. The reason for this development is the application of graphic processing units (GPUs) for massive data parallel tasks besides rendering. Video processing in the context of this paper is defined as part of signal processing, which applies filters and transformation on video sequences and frames [15].

While current video processing frameworks offer the opportunity to program own GPU-based plug-ins for real-time video processing [12] or implement own GPU-based frameworks [16], a developer still has to deal with the constraints of the GPU memory management and its limitations. Therefore, a developer is required to have a deep understanding of the functionality of the GPU to program own real-time filters. Still, for general GPU programming tasks there are frameworks which have a much more suitable level of abstraction, such as the map and reduce framework *MARS* from He et al. [4]. But those focus on general map and reduce tasks and not on video processing itself. Thus, this paper introduces a redefinition of the map and reduce concept, which is more suitable for video processing using implicit GPU programming. This facilitates GPU programming without a deep technical understanding of the underlying hardware with less lines of code. Our approach is a new interpretation of the map and reduce implementation for *implicit parallelism* on cluster based systems presented by Dean and Ghemawat [3].

Moreover, to ensure efficient usage of memory, which is bounded to certain device-specific limits, such as the memory in embedded or mobile devices, this paper introduces the concept of pre-allocated *ringbuffers* for the video data handling. Avoiding dynamic reallocation of memory and inspired by the disruptor implementation in the *LMAX Disruptor* [13]. To summarize, this work makes the following contributions:

1. it presents a concept for bounded and transparent handling of GPU memory for video processing,
2. a novel concept of a map and reduce framework for real-time video processing redefining both map and reduce to be more suitable for video processing and a bounded memory concept,
3. an interoperable implementation of the presented concept based on OpenGL [10].

The remainder of this paper is structured as follows. Section 2 discusses the fundamental ideas of the map and reduce concept as well as implicit parallelism in functional programming for data parallel tasks in video processing. The idea of using a *ringbuffer* for bounded memory allocation and the combination of both concepts into an efficient framework for real-time video processing called *VideoMR* is presented (Section 3). Section 4 briefly describes a prototypical implementation based on OpenGL [10] which is widely used on all platforms including mobile devices. Furthermore, Section 5 demonstrates and explains the framework by means of a move detection filter. Section 6 discusses the performance, code reduction, and limitations of the prototypical implementation (Section 6), followed by the benefits of bounded memory allocation and implicit parallelism for video processing tasks and a generalization of the introduced concept for different types of data streams (Section 7).

2 RELATED WORK

The idea of using *map* and *reduce* in functional programming languages such as Haskell [5] is not new and has been used in large-scale systems for almost a decade. Loidl et al. [6] have shown that an implicit parallelization of functional code in languages such as Glasgow Parallel Haskell (GpH) [14], as extension of Haskell98 [5], can be efficient while reducing programming overhead for the parallelization task.

This concept was further developed by Dean and Ghemawat [3], focusing on a definition of the *map* and *reduce* principle known from the functional paradigm for cluster systems, hiding the details of parallelization, load balancing, and other tasks of cluster-based systems. The computation is therefore divided in a *map* and *reduce* step, which have to be defined by the developer:

$$\begin{array}{rr} map(k_1, v_1) \rightarrow & list(k_2, v_2) \\ reduce(k_2, list(v_2)) \rightarrow & list(v_2) \end{array}$$

The *map* function receives an input *key/value* pair (k_1, v_1) and emits a list of *intermediate key/value* pairs $list(k_2, v_2)$. Those will be sorted into lists of values related to a specific key $(k_2, list(v_2))$ and passed on to a *reduce* function. According to the key, the *reduce* function merges the list of values $list(v_2)$ to a smaller list. The size of the output is typically one or zero. Functions defined this way can be automatically parallelized by the library. Compared to explicit parallel computing models [7], the overhead in programming time has been significantly reduced.

A GPU-based implementation is presented by He et al. [4]. GPUs, originally developed for rendering purposes only, are used since 2002 for general computing tasks [9] as well. The idea behind this is that GPUs perform very well on data parallel tasks, because they have been designed to render high amounts of vertices in parallel. Therefore, they have multiple computation units, which can perform the same task on different data in parallel. Nevertheless, porting *map* and *reduce* to a GPU can still be improved, by taking into account that the dynamic memory model, required for the *intermediate key/value* pairs, is not preferable for GPUs. Furthermore, the specifics of video processing can be used to define an own *map* and *reduce* concept for real-time video processing using the pixel coordinates of each frame as implicit key.

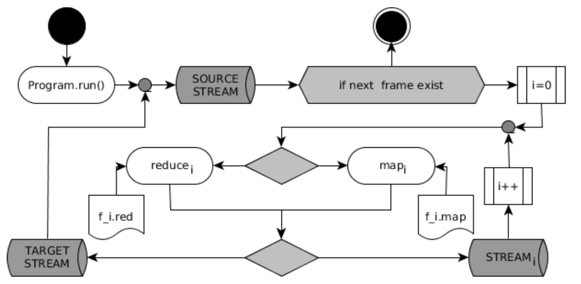


Figure 1: Overview of the work flow of *VideoMR*. As long as frames can be loaded from a source stream they are processed by the map and reduce operations, defined by the corresponding program, and finally pushed to an output stream.

To support a static bounded memory concept, a data structure that can handle massive data is preferred. An efficient approach has been introduced by Thompson et al. with a *ringbuffer* concept called *disruptor* as an alternative for queues in the *LMAX Disruptor* framework [13]. This framework addresses the problems occurring in massive financial data exchange. Because of the *ringbuffer* structure, memory has to be allocated only once and is bounded by the total size of all *ringbuffers*. Moreover, memory can be allocated in advance and reused during processing.

3 CONCEPTUAL OVERVIEW

VideoMR combines the ringbuffer concept for bounded memory allocation and a redefinition of the map and reduce paradigm, to fulfill both: the requirements of modern GPUs and the requirements of real-time video processing. In contrast to existing map and reduce frameworks for GPUs, VideoMR does not focus on transferring map and reduce implementation to the GPUs, but instead on implementing a novel interpretation for video processing.

With respect to both concepts, a *VideoMR program* can be defined using a combination of ringbuffers denoted as *streams*, connected through map and reduce *operations*. A specific *source stream*, e.g., a video loader or generator, passes frames to the first map or reduce operation as long as new frames are available. The cor-

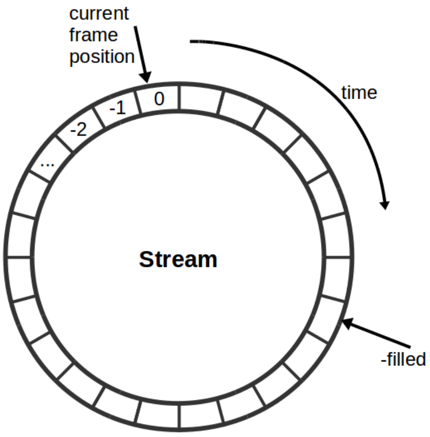


Figure 2: This figure shows the implementation of streams using a ringbuffer. Continuously, the ringbuffer is filled with input video frames. The index indicates the time steps that have passed with respect to the current frame. One pointer is indexing the current frame. Another indicates whether the ringbuffer is full and therefore the next frame will overwrite the oldest one in the stream.

responding operations are processed and executed until new streams are connected. If a target stream, such as a *display* or *writer* stream, is reached, the program restarts for the next frame. This process is summarized in Figure 1 (previous page).

Memory-bounded Streams

The basic data structure, implemented based on the disruptor concept [13], is the *stream*. A stream is a finite sequence of frames. For efficient memory usage for every stream a ring-buffer of k frames is allocated in video memory. Further, a *pointer* to the current frame is initialized and the *counter* for the number of frames in the stream is set to zero. A stream, therefore, represents the current sub-sequence of frames of a video (Figure 2). This temporal information of the streams distinguish video processing from image processing.

Map and Reduce for Video Processing

Initialized *streams* instances can be connected by defining *map* or *reduce* operations (Figure 3) with an input stream s_{in}, and output stream s_{out}, as well as a set of side streams $\{s_0, s_1, \dots, s_{n-1}\}$. Both functions are defined in the same way to be easily combinable:

$$\begin{aligned} map(s_{in}[, s_0, s_1, \dots, s_{n-1}], func.map) &\quad \rightarrow s_{out} \\ reduce(s_{in}[, s_0, s_1, \dots, s_{n-1}], func.red) &\quad \rightarrow s_{out} \end{aligned}$$

The developer can now implement own map and reduce functions by passing definitions in a `func.map` or `func.red` file, denoted as code *snippet*, to the operation. Examples of such code snippets are shown in Section 5.

By implementing these operations, a developer can access the input stream s_{in} and has to write the result to the output stream s_{out}. The side streams $s_0, s_1, \dots, s_{n-1}$ can be used to pass information from earlier processing steps or other constant application data to the specific operation. The concept of a key/value pair of the original map and reduce concept is replaced by the fundamental approach that a pixel position itself serves directly as key for the respective pixel value. Therefore, a key/value pair consists of the tuple $(position\, x, position\, y)$ as the key, and the emitted value would represent the result of an operation on those pixel in time. Still, this approach is not directly comparable to the original idea, because the introduced map and reduce concept is derived from the general concept of map and reduce in functional programming and therefore does not require any keys. However, the general concept, which both approaches share, is that the map function is applied several times to different input data and the reduce function is aggregating those data.

Map Operation Interface

For the *map operation* (Figure 3 and 5), a developer has to define a function that is applied to every pixel of a frame of the output stream s_{out}. The frame is subsequently pushed as first frame to s_{out}. Every stream can be accessed using a *stream structure* representing meta data of the streams such as width, height, and size. To retrieve the position of the pixel to process (serving as key in the operation), the developer has to call $vmr\_getPosition()$.

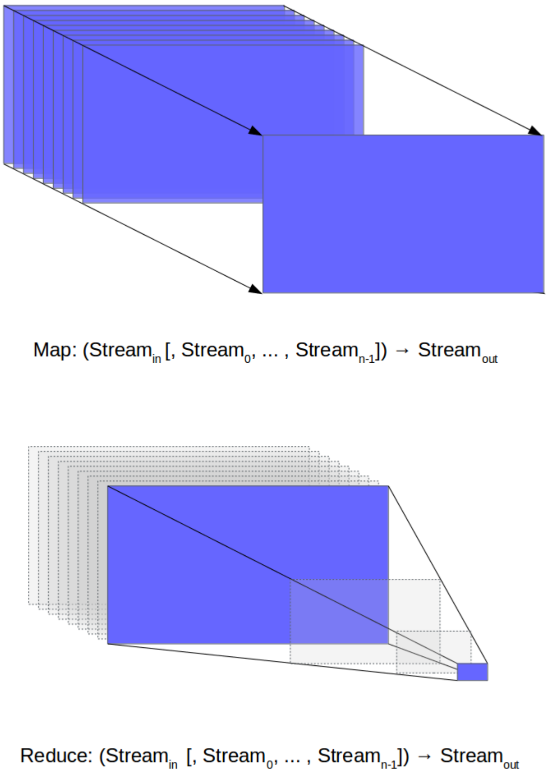
The data at the respective position in the input stream or a side stream can be accessed by the function $vmr\_getStreamDataX(pos,\ time)$. Here, X represents a placeholder for 'In' for s_{in} or the number$i$ of the side stream $(s_0, s_1, \dots, s_i, \dots, s_{n-1})$. The function parameter pos defines the pixel position and $time$ the temporal difference between the frame and the current frame.

To write to an output position, the developer has to call $vmr\_emitToStreamOut(pos,\ pixel)$ explicitly, or, to write to a side stream: $vmr\_emitToStreamX(pos,\ pixel)$ respectively.

Reduce Operation Interface

To implement a *reduce operation* (Figure 3 and 5), the developer is required to define a function that is called for an intermediate frame. This frame is half the width and height of the preceding intermediate frame. During processing, this function is repeatedly called as long as the resulting frame is larger than the least basic resolution of 1×1, or *vmr_emitToStreamOutAndBreak*(*pos*,*pixel*) has been called.

To retrieve the data from a preceding intermediate frame, the developer calls the *vmr_getOldData*(*pos*) function. To emit the resulting pixel information, the function *vmr_emitNewData*(*pos*, *pixel*) has to be called. For an efficient and bounded memory allocation, two intermediate frames are allocated per reduce operation in advanced and used alternated as target or source – so called *ping-pong processing*. Especially for the computation of image or video metrics (e.g., the mean color of a frame), the reduce function is repeated until a minimal frame size of 1×1 is reached. Nevertheless, to reduce the resolution of a frame to a desired size (resolution), the developer can call *vmr_emitToStreamOutAndBreak*(*pos*, *pixel*).



Map: (Stream$_{in}$ [, Stream$_0$, ... , Stream$_{n-1}$]) → Stream$_{out}$

Reduce: (Stream$_{in}$ [, Stream$_0$, ... , Stream$_{n-1}$]) → Stream$_{out}$

Figure 3: Illustration of the general concept of map and reduce operations for video data. The top figure illustrate a map operation that executes its corresponding function for every pixel of the output frame. The bottom figure shows a reduce operation, where a neighborhood of four pixels is merged until the loop is terminated. Afterwards, the resulting frame is emitted to the output stream.

4 IMPLEMENTATION

The presented concept is prototypical implemented using OpenGL [10]. This application programming interface (API) is platform-independent, open source, and supported on most mobile devices. Moreover, OpenGL offers *compute shader*, a generalized interface to GPU programming and is, therefore, the ideal choice for a prototypical implementation.

The stream data structure is implemented using *shader storage buffer objects (SSBOs)*. These have two major advantages: (1) they can be as large as the GPU memory and (2) they are writable at random access. For the implementation of the map and reduce operations compute shader are used. This is a shader type introduced to implement general purpose GPU (GPGPU) operations. We rely on, but are not limited to, the OpenGL Shading Language (GLSL) for operation implementation.

The map and reduce operation are implemented using wrapped compute shader. The discussed API functions are automatically generated for every shader and the streams are automatically passed by the program. The *SSBOs* will be bound to the shader in the beginning of the program run-time.

The basic data structures of VideoMR are implemented using object-oriented design based on GLObjects [1], an object-oriented wrapper for OpenGL. This reduces the code size of the library and helps to extend the framework later on by applying the concepts of the object-orientated paradigm. The framework comprises *core* and *optional* classes (Figure 4), because libraries used for loading or displaying may not be supported or required in every run-time environment. An overview of the functions accessible in a map or reduce operation is shown in Figure 5.

5 APPLICATION EXAMPLE

Discussing a simple move detection program will facilitate an understanding of our implementation of VideoMR and the usage of map and reduce operations for video processing in general.

Assuming a constant camera position, movement in a video can be detected by finding a peak in the first order derivation of the video stream. Thus, a discrete approximation of the derivation is the difference of a pixel over time. If this difference is larger than the mean color of the current frame, movement can be assumed, otherwise not. To reduce noise, this movement has to be detected in at least two subsequent frames at the same position.

Listing 1: Map and reduce main program.

```
// init program
auto prog = std::make_shared<vmr::GlfwProgram
    >();
```

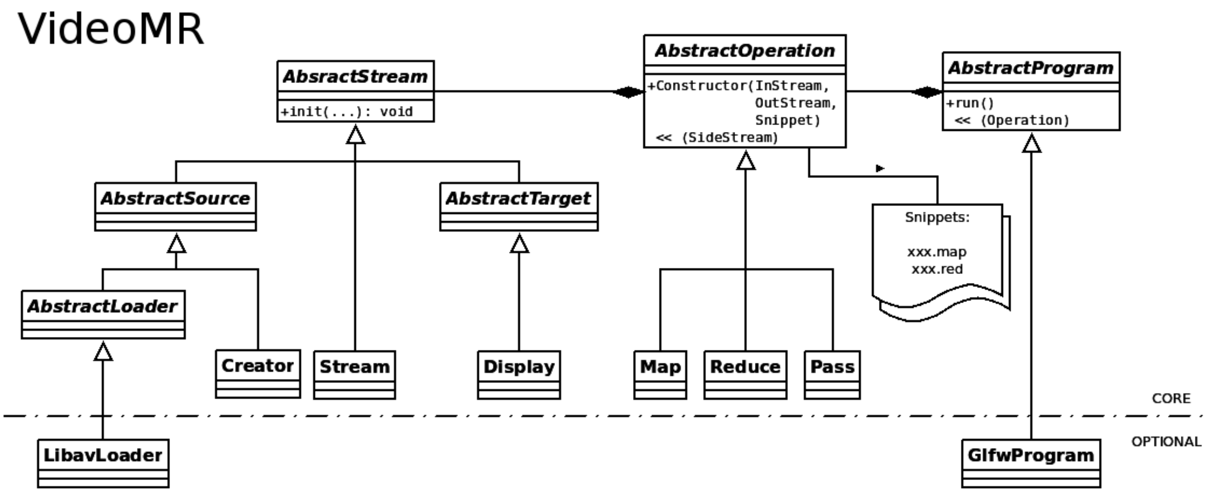


Figure 4: The overview shows the Unified Modeling Language (UML) class structure of the VideoMR framework. Every abstract operation is defined on input, output, and side streams. The snippets contain the source code for a particular operation. Therefore, a program contains multiple operations.

```
// init video loader stream
auto source = std::make_shared<vmr::
    LibavLoader>("./example.mov");
// set the size of the stream to 3
source->init(3);

// mean
// init the result stream for the mean,
// without an explicit init call
// the size will be the size of the
// input stream of the operation
auto mean = std::make_shared<vmr::Stream>();
// init the mean reduce operation and
// add the source stream as input and
// the mean stream as output
auto meanRed = std::make_shared<vmr::Reduce>(
    source,mean,"./mean.red");

// move detection
// init the result stream for the move
// detection, without an explicit init call
// the size will be the size of the
// input stream of the operation
auto move = std::make_shared<vmr::Display>();
// init the move map operation and add
// the source stream as input and the
// move stream as output
auto moveMap = std::make_shared<vmr::Map>(
    source,move,"./move.map");
// add the mean as side stream
*moveMap<<mean;

// setup program and add the operations to it
*prog<< meanRed
     << moveMap;

// run program
prog->run();
```

The code shown in Listing 1 initializes the program. Afterwards, a video loader is initialized and connected with a *reduce* operation to compute the average color of each frame. The output stream of this operation is, together with the loader, connected to a *map* operation computing the actual movement detection, receiving the mean as side stream. The resulting stream is displayed subsequently. Both operations, *map* and *reduce*, receive a file with the concrete implementation as argument.

Listing 2: Exemplary implementation of a mean value computation (mean.red)

```
// get position of current thread
ivec2 pos = vmr_getPosition()*2;

// get four pixel neighbourhood
vec3 c = vmr_getOldData(pos);
pos.x += 1;
c += vmr_getOldData(pos);
pos.y += 1;
c += vmr_getOldData(pos);
pos.x -= 1;
c += vmr_getOldData(pos);
c/=4;

// write result to current position
vmr_emitNewData(vmr_getPosition(), c);
```

The details of the *reduce* operation file is shown in Listing 2. In every reduce step, a four pixel neighborhood is summarized and divided by four. This computes the local average until only one pixel is left, which then serves as the global mean. If the intermediate size equals a frame resolution resolution of one, the result is automatically pushed to the output stream.

Listing 3: move.map

```
// get position of current thread
ivec2 pos = vmr_getPosition();

// get data from last three frames
vec3 c;
```

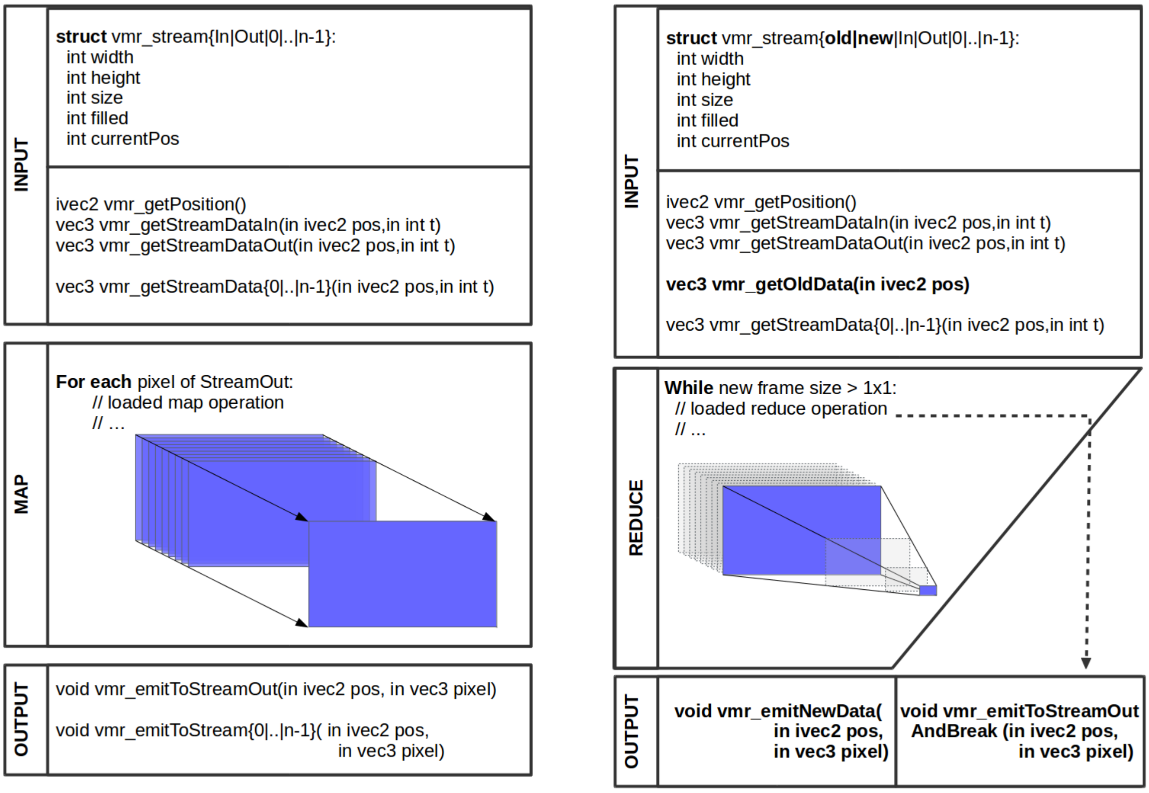

INPUT
struct vmr_stream{In|Out|0|..|n-1}:
int width
int height
int size
int filled
int currentPos
ivec2 vmr_getPosition()
vec3 vmr_getStreamDataIn(in ivec2 pos,in int t)
vec3 vmr_getStreamDataOut(in ivec2 pos,in int t)
vec3 vmr_getStreamData{0|..|n-1}(in ivec2 pos,in int t)
MAP
For each pixel of StreamOut:
// loaded map operation
// ...
OUTPUT
void vmr_emitToStreamOut(in ivec2 pos, in vec3 pixel)
void vmr_emitToStream{0|..|n-1}(in ivec2 pos, in vec3 pixel)
INPUT
struct vmr_stream{old|new|In|Out|0|..|n-1}:
int width
int height
int size
int filled
int currentPos
ivec2 vmr_getPosition()
vec3 vmr_getStreamDataIn(in ivec2 pos,in int t)
vec3 vmr_getStreamDataOut(in ivec2 pos,in int t)
vec3 vmr_getOldData(in ivec2 pos)
vec3 vmr_getStreamData{0|..|n-1}(in ivec2 pos,in int t)
REDUCE
While new frame size > 1x1:
// loaded reduce operation
// ...
OUTPUT
void vmr_emitNewData(in ivec2 pos, in vec3 pixel)
void vmr_emitToStreamOut AndBreak (in ivec2 pos, in vec3 pixel)

Figure 5: Overview of map and reduce operations and their interfaces. The left figure illustrates how to program an own map operation, access the input data, and emit to the output stream. The right figure explains the same sequence for the reduce operation.

```
vec3 c1 = vmr_getStreamDataIn(pos,  0);
vec3 c2 = vmr_getStreamDataIn(pos, -1);
vec3 c3 = vmr_getStreamDataIn(pos, -2);

// compute difference
float diff1 = abs(c1.r-c2.r)+abs(c1.g-c2.g)+
      abs(c1.b-c2.b);
float diff2 = abs(c2.r-c3.r)+abs(c2.g-c3.g)+
      abs(c2.b-c3.b);

// get mean from side stream
ivec2 pos2 = ivec2 (0,0);
vec3 meanC = vmr_getStreamData0(pos2,0);
float aveMean=(meanC.r+meanC.g+meanC.b)/3;

// compare with mean
if (diff1>aveMean && diff2>aveMean){
    c = vec3(255, 255, 255);
} else {
    c = vec3(0, 0, 0);
}

// write result to current position
vmr_emitToStreamOut(pos,c);
```

The algorithm for the move detection used in the *map* operation shown in Listing 3 is a special version of *differential images*, described by Collins et al. [2]. First, the pixel value of the current frame and the two preceding frames are computed. Afterwards, the difference between the first and the second frame as well as the difference between the second and the third frame is computed. If both differences are larger than the mean color of the step before, the emitted pixel is set to white, otherwise to black. The basic idea is, that noise in two different frames is visible in two different positions and therefore filtered using this computation.

The result of this operation for a single frame of a video of an animated fractal set is shown in Figure 6. While the right side of the figure shows the frame source, the left part shows the movement-detection filter applied. Both, the moving borders and also the details of the fast changing inner structure of the fractal can be detected.

6 RESULTS AND DISCUSSION

The prototypical implementation of VideoMR and the example shown in Section 5 demonstrate that efficient GPU programming is possible without explicitly taking care of memory handling or parallelization using the presented map and reduce concept.

Code Reduction

Furthermore, the code can be reduced in comparison with a pure implementation using GLobjects

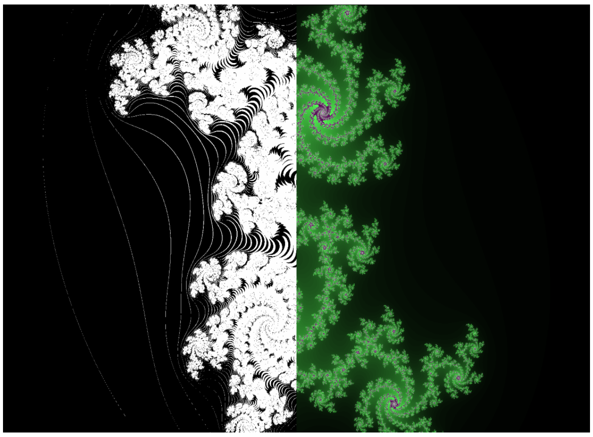


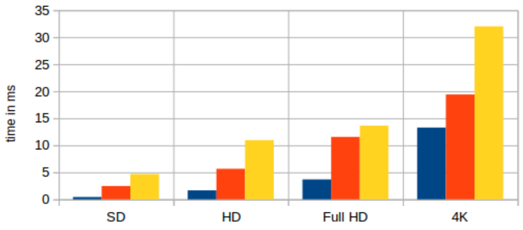
Figure 6: This frame shows on the right side an animated fractal set and on the left side the move detection filter applied for that fractal.

and OpenGL. Referring to the movement detection example in Section 5, the memory management and program setup requires 392 lines of code, the reduce operation 109, and map operation 70 (571 in total). In this example, less than 50 lines are required using VideoMR, a reduction by a factor of 10.

Performance Evaluation

The performance of the prototypical implementation has been computed for four different resolutions: SD (720 × 576), HD (1280 × 720), Full HD (1920 × 1080), and 4K (3840 × 2160). Figure 7 shows the runtime performance according to these resolutions for a map and reduce program using a single source and target stream with a single map operation, a single reduce operation, as well as a complex example with two map and a single reduce operation.

In video processing, real-time processing can be defined by achieving more than 24 frames-per-second (FPS). This is approximately the sampling rate of the human eye, i.e., maximal 40 ms per frame. Therefore, real-time processing can be achieved on a Quadro K1000M GPU total (dedicated) video memory 2048 MB in the complex example running at 4K resolution (Figure 7). Still – if not required for displaying the result – the OpenGL rendering context can slow the processing down.



	SD	HD	Full HD	4K
🟦 map	0.4426 ms	1.6692 ms	3.6844 ms	13.2744 ms
🟥 reduce	2.4658 ms	5.6576 ms	11.563 ms	19.4006 ms
🟨 complex	4.6846 ms	10.958 ms	13.6282 ms	31.9996 ms

Figure 7: Performance for different video resolutions in milliseconds. Displayed are different resolutions, applied to a single map operation, a single reduce operation, and a complex example containing two map and a single reduce operation.

Memory Boundaries

To compute the GPU video-memory usage the following equations can be used:

$$\begin{aligned} usage(Stream) &= Stream.size \cdot Stream.width \\ &\quad \cdot Stream.height \cdot 3Byte \\ usage(Map) &= (count(Streams) \cdot 5 + 2) \cdot 4Byte \\ usage(Reduce) &= (count(Streams) \cdot 5 + 4) \cdot 4Byte \\ &\quad + Stream_{in}.width \cdot Stream_{in}.height \\ &\quad \cdot 3Byte \cdot 2 \end{aligned}$$

Limitations and Improvements

However, some operations in video processing are not parallelizable and therefore not programmable with map and reduce. These classes of problems can be approached within VideoMR by introducing a specific implementation of an explicit operation that copies the data transparently to the main memory and can then be programmed with a serial approach.

Also the concept of frames as an array of red, green, and blue values is a limitation for modern 2.5D or 3D video data. Thus, in future work a general concept of n-dimensional buffers instead of frames will be used. Frames then can be handled defining a $frame[width][height]$ as a $buffer[width][height][3]$. This makes it also possible to handle keyboard input or the sound of a video and other data as streams.

Moreover, the current limitation to $8Bit$ values per channel is not preferable for using other data than images or videos. Thus, the extension with template classes to decide which main data type to use should be added in future work. Still OpenCL [11] and CUDA [8] are rarely supported in embedded and mobile environments, but future implementation should also consider using them, because they are not depending on a render context that possibly impacts runtime performance, which is, for read and write operations, not required.

While this paper shows the suitability for real-time video processing, comparison with other map and reduce frameworks will be part of future research. The reason for that is that current benchmarks for map and reduce frameworks are not focusing on video processing tasks and therefore a suitable one has to be developed before.

7 CONCLUSION

This paper presents a concept for transferring existing map and reduce processing metaphors to the domain of video processing using GPUs. It describes a prototypical implementation based on OpenGL and the OpenGL Shading Language. This proof-of-concept demonstrates the efficiency of map and reduce for modern real-time video processing applications. Implementations can be performed using less lines of code with transparent memory handling. Furthermore, the disruptor concept of ringbuffers offers the opportunity to implement video processing on systems such as mobile devices, where memory is a limited resource or the access is restricted by the operating system itself.

To summarize, current video processing frameworks such as *Gstreamer* [12] focus on transparently using video filters and effects. This enables developers to use existing filters, but gives less support for programming own filters based on GPU programming languages. In contrast, *VideoMR* focuses on a programming paradigm based on redefined *map* and *reduce* strategies, allowing to develop own filters for GPUs that are fast to write using implicit parallelism concepts designed for video processing. A lightweight implementation with *OpenGL* [10] ensures the portability and interoperability with existing frameworks. Moreover, the bounded memory handling fulfills the requirements of mobile development with limited or bounded memory resources.

8 REFERENCES

[1] GLObjects. `https://github.com/hpicgs/globjects`. Accessed: 2015-02-04.

[2] Robert Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, and Osamu Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Pittsburgh, PA, May 2000.

[3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.

[4] Bingsheng He, Wenbin Fang, Qiong Luo, Naga K. Govindaraju, and Tuyong Wang. Mars: A mapreduce framework on graphics processors. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, PACT '08, pages 260–269, New York, NY, USA, 2008. ACM.

[5] Simon P. Jones, John Hughes, and Lennart Augustsson. *Haskell 98: A Non-strict, Purely Functional Language*. 1999.

[6] H.-W. Loidl, F. Rubio, N. Scaife, K. Hammond, S. Horiguchi, U. Klusik, R. Loogen, G. J. Michaelson, R. Peña, S. Priebe, Á J. Rebón, and P. W. Trinder. Comparing parallel functional languages: Programming and performance. *Higher Order Symbol. Comput.*, 16(3):203–251, September 2003.

[7] Kato Mivule, Benjamin Harvey, Crystal Cobb, and Hoda El-Sayed. A review of cuda, mapreduce, and pthreads parallel computing models. *CoRR*, abs/1410.4453, 2014.

[8] NVIDIA. *NVIDIA CUDA Programming Guide*. NVIDIA, The address of the publisher, 2.3 edition, 2009.

[9] Matt Pharr. *Part IV - General-Purpose Computation on GPUs: A Primer*. GPU Gems 2. Addison-Wesley Publishing Company, 2005.

[10] M. Segal and K Akeley. *The OpenGL Graphics System: A Specification*. Silicon Graphics Inc., 4.4 edition, 2014.

[11] John E. Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *IEEE Des. Test*, 12(3):66–73, May 2010.

[12] W. Taymans, Baker S., A. Wingo, S. Bultje, and S. Kost. *GStreamer Manual*, 2014.

[13] M. Thompson, D. Farley, M. Barker, P. Gee, and A. Stewart. *DISRUPTOR: High performance alternative to bounded queues for exchanging data between concurrent threads*. LMAX, 2011.

[14] Philip W. Trinder, Kevin Hammond, Hans-Wolfgang Loidl, and Simon L. Peyton Jones. Algorithm + Strategy = Parallelism. *Journal of Functional Programming*, 8(1):23–60, January 1998.

[15] Yao Wang, Joern Ostermann, and Ya-Qin Zhang. *Video Processing and Communication*. Prentice Hall, 2002.

[16] Rubin Xu. A gpu-enabled real-time video processing library, part ii, computer science tripos, trinity college, 2010.

ACKNOWLEDGEMENTS

This work was funded by the German Federal Ministry of Education and Research (BMBF), as part of the InnoProfile Transfer research group "4D-nD-Vis" (`www.4dndvis.de`).

Perspective Correction of Panoramic Images created by Parallel Motion Stitching

João Gonçalves
Fraunhofer AICOS
Rua Alfredo Allen 455
4200-135 Porto,
PORTUGAL
joao.goncalves@fraunhofer.pt

David Ribeiro
Fraunhofer AICOS
Rua Alfredo Allen 455
4200-135 Porto,
PORTUGAL
david.ribeiro@fraunhofer.pt

Filipe Soares
Fraunhofer AICOS
Rua Alfredo Allen 455
4200-135 Porto,
PORTUGAL
filipe.soares@fraunhofer.pt

ABSTRACT

This paper deals with the problem of correcting the perspective distortion in panoramic images created by parallel motion stitching. The distortion is revealed by lines that appear to converge at the infinity, but are actually parallel. A camera cart shoots from multi-viewpoints aiming a parallel motion to the scene that is photographed. The perspective effect arises on panoramas while stitching several images taken from the camera, slightly panning in both directions between shots along the motion path. In this paper, we propose a solution to handle different camera translation motions and be able to stitch together images with a high-level of similarity, also having repetition patterns along a vast continuity of elements belonging to the scene. The experimental tests were performed with real data obtained from supermarket shelves, with the goal of maintaining the correct amount of product items on the resulting panorama. After applying the perspective correction in the input images, to reduce cumulative registration errors during stitching, it is possible to extract more information about the similarity between consecutive images so that matching mistakes are minimized.

Keywords

Affine transformation, panorama, stitching, parallel motion, multi-viewpoint, similarity, retail.

1 INTRODUCTION

Image stitching can be a complex sequence of image processing steps, especially when considering stitching several high resolution images photographed with wide-angle or fisheye lenses, at close range from the scene. Every time a high-level of similarity occurs between a pair of images, and a repetition pattern exists with a vast continuity of elements belonging to the scene, the errors in the final panorama rapidly rise with the number of pictures to blend. This is particularly important in scenarios where the exact number of elements of the reality should remain in the captured panorama.

For retailers, keeping supermarket shelves stocked is a vital part of running a successful operation. Monitoring shelves have always been an expensive and inefficient manual process, requiring stock clerks to do it throughout the day. The aforementioned stitching problem gets worse if one needs to capture the shelves of a long aisle of a supermarket. There is a limited distance between two opposite shelves in an aisle. The approach for a single photograph from the opposite side would either, capture a short portion of the shelf or have distortion towards the edges for wider field-of-views (using a fisheye lens for instance). Stitching errors must be avoided because one shelf has a planned number of items of a given product, and the final panorama must have the same number of items for control.

The problems described are not exclusive to supermarkets. In general, single-perspective photographs are not very effective at conveying long and roughly planar scenes, such as a river bank or the facades of buildings along a street. In those cases taking a photograph from a faraway viewpoint could be an alternative, but usually it is not possible to get far enough in a dense city to capture such a photograph. Even if it was possible the result would lose the track of perspective depth.

The stitching technique proposed in this work should not be confused with classic stitching around a view point. Rather than relying on a single projection, Kopf et al. [Kop09] introduce locally-adapted projections, where the projection changes continuously across the field-of-view. They also employ a perspective correction while ensuring continuous transitions between captured regions, thus preserving the overall panoramic context similar to how humans perceive the scene. However, the user must specify where the lines are straight. In addition, Anguelov et al. [Ang10] use a more complex image acquisition with multiple cameras calibrated among each other, on a street view scenario which has distinct constrains from a feature-based homography. The resulting 360° panoramas do not provide a good visual perception of a larger aggregate such as a whole city block, as discussed in [Kop10]. Agarwala et al. [Aga06] introduced an approach of producing panoramas that displays these types of long scenes. However, the user must specify the dominant plan of the scene and do not take into account the similarity and repetition of elements in the photos, as the case of products in supermarket shelves that should not appear merged due to stitching errors. In addition, the camera is manually moved and positioned along the scene.

The present paper departs from [Aga06] by adding motion to the multi-viewpoint stitching problem, to build a parallel motion stitching of photographed shelves, fully automatically. A camera cart shoots from multi-viewpoints and a slight motion of the camera as a very significant impact on the final panorama. When the camera moves in parallel to the scene that is being photographed, the parallel field-of-view is correct from the center of one image to the center of the next image, but it is not correct to the final image plane that appears in perspective projection, i.e. the last image is in perspective projection in relation to the first image. In particular, we focus on auto-calibration on a single photo. It is known that the final convergence of multi-image stitching depends (like any problem of optimization) how close the initial parameters are to the optimal solution. We aim to build a high resolution and long panorama and, therefore, we need a method that makes all images globally consistent without major artifacts.

2 METHODOLOGIES

In this paper, a stitching pipeline is proposed to essentially combine the orientation information extracted from parallel lines in the images with perspective effect, with the epipolar geometry to remove artifacts on the panoramic images caused by the semi-translational motion (translation plus rotation) of the camera that is panned relative to the scene. Any orientation changes are even more prominent with fisheye lenses.

The novelty of the paper lies in: preprocessing the images individually to remove the affine transformation (perspective) using position and angle from parallel lines present on image; roughly compute the region with high similarity between consecutive images; and as a last step, iterative matching process optimized by the homography angle along a specific plan.

The present methodology aims to reduce the accumulation of errors in registration process (homography-based stitching) from stitching pipeline presented by [Bro07] which is implemented on OpenCV [OpenCV]. Brown and Lowe 2007 extract features from all images. Then using those pairs of features, a homography matrix is calculated. This homography matrix is then used to warp one of the images onto other one, correcting local matching mistakes by a global regularization of images without a defined order. In this work in which the images follow an order, these kind of mistakes are solved even before the global regularization step, making the initial guess to be closer to the final global homography estimation.

We aim to overcome some problems of feature-based approaches, as images with lack of texture or similarity of features, which in our case is very common. Since all their camera spaces lie in the same horizontal plane in world space, only the translation plane from the camera space are collinear between all shots and coplanar with the translation plane from world space. Moreover, the different camera spaces are roughly aligned with world space which is not enough to turn the final panorama flat and rectilinear.

Figure 1 presents the proposed stitching pipeline. First the photos with a fisheye lens are acquired in motion (see Figure 2). Then, compute homography based on information from the parallel lines and apply affine transformation to the input images. After this it is possible to find the most similar region between consecutive images, so that matching mistakes are avoided. Moreover we improve the RANSAC (Random Sample Consensus) [Fis81] convergence (less iterations) by using this information. After this preprocessing stages we are ready to finish the registration and composing the images that are taken from parallel motion. Further details on the stitching pipelines are presented in the next sections.

2.1 Defisheye

Fisheye lenses create hemispherical images that must be undistorted to create a linear panorama. To accomplish this we follow Devernay and Faugeras [Dev95a] approach. The process from distorted to undistorted the fisheye effect is reversed so that the correction of the whole images is a lot faster. From the radius of the undistorted image $\mathbf{r_u}$ (the distance in pixels from the center of the image) (1), the correspondent radius of the distorted image $\mathbf{r_d}$ (2) is calculated where **f** is the apparent focal length of the fisheye lens, which is not necessarily the actual focal length of the fisheye. Basically, for each pixel in the processed image the corresponding pixel is determined in the distorted image.

$$r_u = \sqrt{x^2 + y^2} \tag{1}$$

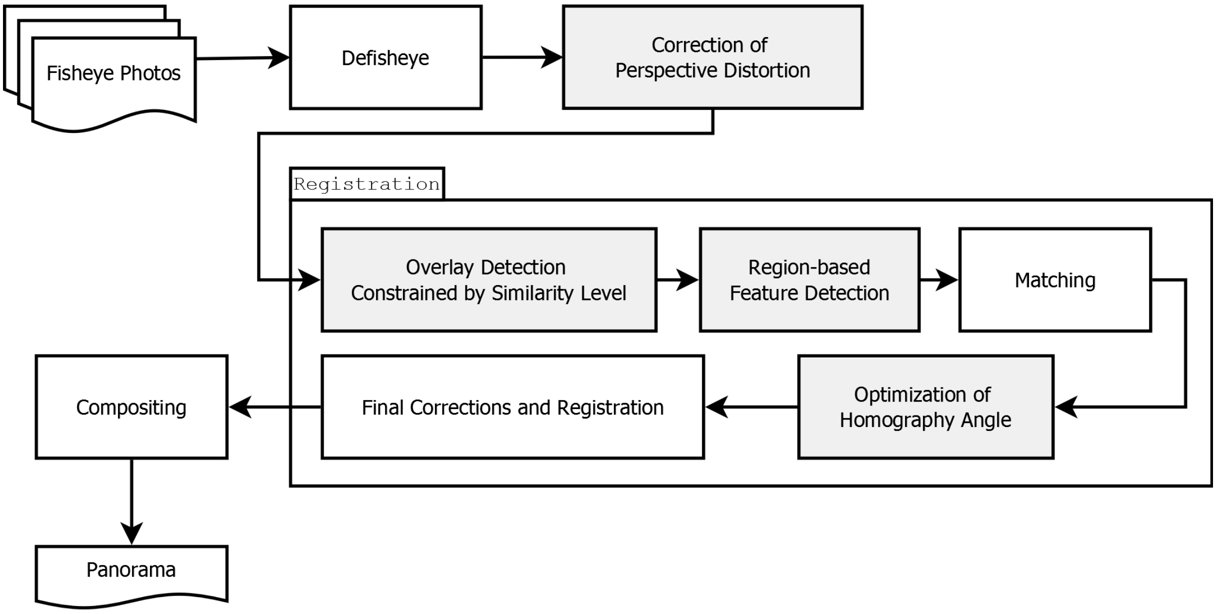

Fisheye Photos
Defisheye
Correction of Perspective Distortion
Registration
Overlay Detection Constrained by Similarity Level
Region-based Feature Detection
Matching
Optimization of Homography Angle
Final Corrections and Registration
Compositing
Panorama

Figure 1: Stitching Pipeline. The main contributions are highlighted in gray.

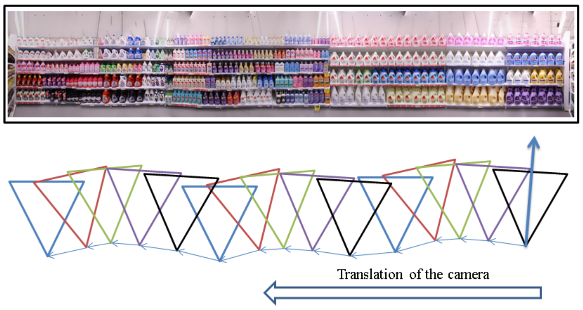


Figure 2: Example of scene to photograph.

$$r_d = f \arctan \frac{r_u}{f} \qquad (2)$$

2.2 Correction of Perspective Distortion

Straight lines are common in man made environments, especially in supermarket aisles. Devernay and Faugeras [Dev01b] introduced the idea of "Straight lines have to be straight" to calibrate intrinsic camera parameters for a set of cameras. A similar approach is employed here, to compute the affine transformation of the image and posterior warp, but making the orthogonal plane from camera space roughly aligned between shots along the motion path of a camera cart. From images with parallel lines we can compute the orientation of the camera with respect to the scene. Considering a set of coplanar parallel lines in 3D world space like the ones in Figure 3. The lines in the image appear to converge when the camera is panned. Due to that, we search lines where the convergence is higher in the top and bottom of the images (lines closer to the center of the camera remain more straight) where it is more probable to find a line with the biggest slope. Using a line detector algorithm, we search for these lines (shelves) and calculate the slope that makes them converging, and then warp the image so they do not appear to be converging.

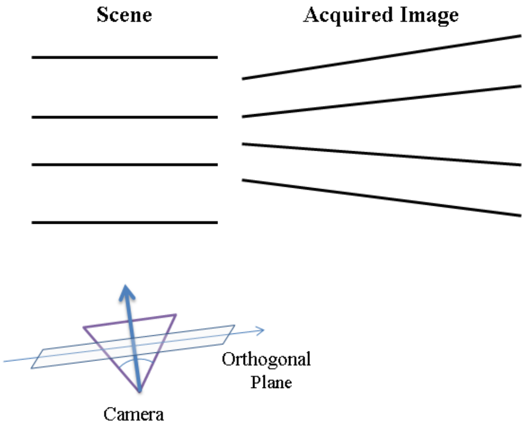


Figure 3: Example of panning camera. Left: the scene to be captured. Right: the image obtained due to the orthogonal plane not being parallel to the scene.

The perspective distortion is revealed by lines that in camera appear to converge at the infinity, but are actually parallel. This effect is based primarily on the panning and distance from the camera to the scene and, to some extent, on the focal length of the lens. It appears exaggerated with a fisheye lens because a bigger field-of-view is captured. In case the camera is panned, one half of the image is covering more area in 3D space than the other half, resulting in the convergence of lines in the photo (see Figure 3).

The classical Hough [Dud72] transform provides a powerful and robust technique for detecting these lines, other curves or predefined shapes in a binary image. Our objective is to locate nearly linear arrangements of disconnected white spots (most probably "broken" lines). Considering that a straight line in the input image is defined by the equation (3) (polar coordinates), where

$$h : (x, y) \rightarrow \rho = x * cos(\theta) + y * sin(\theta) \qquad (3)$$

ρ and θ are two unknown parameters whose values have to be found. Every point in the binary image has a corresponding ρ and θ. Actually, these points form a sinusoidal curve in (ρ,θ) space called Hough Transform (HT) image. The whole HT image form a multitude of overlapping sinusoids that may converge on the same spot. The (ρ,θ) address of such point indicates the slope θ and position ρ of a straight line. This information is used to estimate the homography on a single image.

The scene presented in Figure 3 contains more than one possible line. Even if we divide the image on two halfs we can estimate the address of two lines (four points: two on the top half and two on the bottom half). To compute the homography for correcting the affine transformation on the image, eight points are needed: four points that define the correct image and four points to define the warp image.

2.3 Registration

Image registration is the process to assign the different coordinate system of every image that compose the final panorama, to a single and unique coordinate system. Stitching by feature-based homography requires registration to determine the overlapping region, i.e. to compute the homographies from one image to the subsequent. The key problem in image registration is to find a conversely relation from one image to another especially on perspective image. So first we need to know the coordinate system for every consecutive pair of images. For that task we use the Oriented FAST and Rotated BRIEF (ORB) [Rub11] to find features in every image that compose the panorama. The ORB descriptors are computed from those features, aiming a description of a point that is unique as possible, and matching the most similar descriptions according to Lowe et al. [Low04] to form a panorama, from which the information is extracted from the set of overlapping regions. That information is necessary to estimate the final corrections and obtain the resulting estimated homography (global coordinate system).

2.3.1 *Overlay Detection*

After the previous step of correcting perspective projection, the consecutive images should be aligned. Using the L2-norm, we compute the regions that are more similar inside the overlap formed between each pair of consecutive images. This step is taken in conjunction with a vertical region-based feature detection (detailed in the next section), as we pretend to avoid matching mistakes influenced by a high repetition of products on supermarket shelves. A mask of descriptors can be created knowing the position of this region on the image, which means that we discard the matches that are outside of this region. In case the estimated homography from this region are very far apart from the basic understanding of the scene, this region can slide.

2.3.2 *Region-based feature detection*

In stitching by feature-based homography on which image matching is to be performed, there are basic requirements that interesting points should have:

i clear and mathematically well-founded definition (estimation of descriptors)

ii well defined position on image space (detection of features)

iii local image structures around the interest point, that are mostly unique if it were possible

iv stability for a reliable computation of a interest point under image deformations and illumination variations (invariance across view points).

To estimate reliably the initial guess homography, the intra-image features must be distinctive from anything else in the same image (i,ii and iii). Since objects are likely to repeat, these invariant features can also appear repeated in the scene, and finding corresponding image points can be a huge challenge.

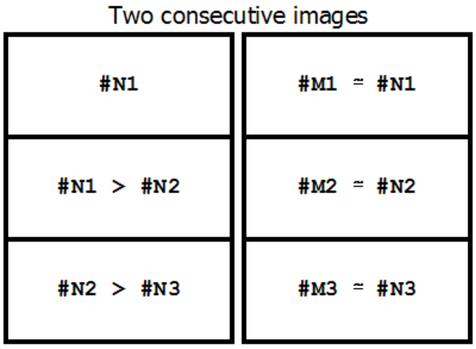


Figure 4: Diagram to extract different cardinality of features in the vertical direction of each image. N and M are detected features.

Knowing the most similar region already detected between consecutive images (horizontal), and because we are photographing shelves with a fixed scheme, it is possible to avoid matching mistakes by extracting different cardinality of features in the vertical direction.

From top to bottom, the image is split into sections (see Figure 4). Similar features in the same image have different distance between neighbor features (since the cardinality is different), to improve the posterior matching process in the stitching pipeline.

2.3.3 Matching

The matching process consists in finding the image features that appear in other image with the same descriptors. In this work, the ORB descriptor is used to locate points of interest and the description of those points. Without the previous steps the number of ambiguity between matches is huge, partly due to the characteristics of the supermarket shelves. On the other hand, the stitching pipeline used in [Bro07] find this ambiguity after processing the initial guess resolving the autocalibration problem (homography between consecutive images). Brown and Lowe, first do the features matching based on local description (nearest neighbor strategy) and label them with multiple matches as ambiguous. Then, in a second stage the ambiguous matches are removed imposing globally consistence constraint which is solved using the Levenberg-Marquardt algorithm.

2.3.4 Optimization of homography angle

To compute a reliable global homography, it is necessary that the initial guess is roughly estimated and close to the optimal solution which implies that the matching process do not have ambiguities, i.e., the images would not have similar structures or repeated patterns. Since this is not possible in supermarket shelves, by using the proposed methodology the first guesses are well estimated. Homographies that describe the affine transformation between pairs of images are estimated using RANSAC estimation method (the outliers of corresponding points can be removed more easily by the RANSAC), based on Singular Value Decomposition (SVD) to extract the rotations and translations matrix out of the estimated homography as explained in [Zis04].

The angle in the orthogonal plane from the rotation matrix is used as a measure for the amount of misregistration (as we move in parallel to the scene, the final stitching should be flat and rectilinear). If the resulting angle is above a threshold the descriptor masks are slided and the matching process is run again with different features. More detail on homography decomposition for extraction of Euler angles can be found in [Mal07].

2.4 Compositing

This stage follows the steps of the stitching pipeline in [Bro07]. A portion of different images is cut and pasted into the final panorama. Naturally, the simple cut and past step leaves artificial edges in the overlapping regions due the difference in camera gain and scene illumination.

Once we have registered all of the input images with respect to each other, we need to decide how to produce the final panorama image. This involves selecting a final compositing surface (flat, cylindrical, spherical, etc.) and view (reference image). It also involves selecting which pixels contribute to the final composition and how to optimally blend these pixels to minimize visible seams, blur, and ghosting [Sze11].

A common approach in warp several images to compose the final panorama is to use one image as reference and warp the others to the reference. With this approach the warp to the last image is the accumulation of all warps, which means that if the referenced image are in perspective projection the second image to be warped must continue the perspective projection in order to respect the global homography. The sequence of successive warps makes the final panorama appear shrinking or with expansion. This is the case when one tries to stitch images taken from parallel motion. For images that have large rotations it can not be handled. Our approach tries to reduce this effects on final panorama.

3 RESULTS AND DISCUSSION

The collection of images for the experimental test was obtained at a large supermarket of Sonae MCH, Portugal. The dataset consists of ten aisles with length varying between 6m and 13m, captured one meter from the supermarket shelves. The setup is focused on creating a high-quality panorama of complete grocery store aisle (see Figure 2). Due to the constructions of grocery stores the distance between aisles is too short to take a photo of the full aisle in length. The solution is to move the camera along the aisle and take several photos, stitching them at the end. Problems do arise from this approach: short distance implies using wide-angle or fisheye lens to capture the full height of aisle. The motion of the camera is not linear and it may appear panned relatively to the previous capture.

Spizhevoy and Eruhimov 2012 [Spi12] do not use pattern-based calibration. The auto-calibration of the camera is done using the epipolar geometry only, assuming that the camera mainly has rotation between viewpoints. Moreover, the work in [Spi12] has stronger constraints with respect to the minimal distance to scene, that has to be twice the value employed here. Contrarily, our method does not require knowledge of the intrinsic camera parameters.

Perfect final panorama implies that all the photos are taken exactly parallel to the aisle, but since the camera is in motion it is more difficult to maintain it perfectly aligned. This means that the photos need to have a large overlapping region (typically at least 50%). Due to the

Algorithm 1: STITCHING IMAGES FROM PARALLEL MOTION

```
Input: A sequence of $n$ ordered images
for $i = 1 \rightarrow n$ do
    1. Undistort fisheye effect.
    2. Compute hough space for top and bottom of
       image and extract the biggest line in these two
       regions.
    3. Using the slope from the previous lines, compute
       homography and warp the image to undo the
       convergence of lines.
end
for $j = 2 \rightarrow n$ do
    4. Compute mask for high similar region between
       consecutive images using norm L2.
    5. Extract ORB features for image $j$ respecting
       cardinality constraint.
    while $angle > threshold$ do
        6. Find k nearest-neighbours for consecutive
           features.
        7. Find geometrically consistent feature
           matches using RANSAC, in combination
           with masks calculated before, to solve the
           homography between images pairs.
        8. Verify image matches using the angle in the
           orthogonal plane, if false slid masks.
    end
end
9. Perform bundle adjustment to solve the rotation $\theta_1$
   $\theta_2$ $\theta_3$[Mal07] and focal length for all cameras.
10. Render panorama using multi-band blending.
Output: Panorama image
```

nature of supermarket aisles, many times the products appear repeated continuously, so the probability of having similarity between two consecutive images is high.

To validate the quality of the correction method proposed herein, we compare two measures that reflect how far from linear a given panorama has become: the percentage difference between the height in the beginning and the end of the resulting panorama, as % Difference Height (DH); the global slope of the aisle perspective distortion, as % Global Slope (GS). The results are summarized in Table 1 that reflects the improvements on different aisle panoramas. The best results obtained with [Bro07] (Before) are compared with our approach to the problem (After). The improvements noted by lower values of DH or GS occur mainly because of the perspective correction, overlay detection method and homography angle refinement. Also, a relevant step is the extraction of different cardinality of features in the vertical direction in the individual images.

Figure 5 and 6 show the gain in visual quality, not only in % GS that represents the degree of perspective effect, but also the DH that shows how misestimated the panorama is compared to the real scene. From Figure 7 to 8, we can visualize how the missing of product items was recovered. As observed in Figure 9, the most perceptible artifact in the final image is the parallax effect in rear products. This parallax level is acceptable for our goal, since only the front facings of the products count. To improve this kind of artifact an alternative would be to follow Zheng et al. 2011 [Zhe11]. The authors construct panoramas from multi-view points using epipolar geometry to remove the perspective effect on final panorama, making use of optical flow to avoid parallax effects. This method requires video capture which has a lot more information to compute a reliable epipolar geometry.

4 CONCLUSION

Parallel motion stitching is definitely an interesting technique for panorama production. This article proposes a stitching pipeline for motion photography, using perspective correction, similarity-constrained overlay detection and homography angle optimization. The results allow us to conclude that it is possible to correct major artifacts in stitched images of supermarket shelves, that usually require a high level of visual quality in the resulting panorama. It is important that the final image avoids a lack of elements in the scene compared to the reality, if possible resembling a unique capture of the whole scene. The panoramas exhibit detailed information even with a photo shooting at multi-view points, without worries on limited spaces, and also during motion. The goal proposed was accomplished without manual intervention on capture nor image processing.

Table 1: Correction quality in parallel motion stitching measured by % Difference Height (DH) and % Global Slope (GS). Before: The best results obtained with [Bro07]. After: Our approach. Lower is better.

	Before		After	
	%DH	%GS	%DH	%GS
Aisle 1	50.45	5.35	6.02	1.43
Aisle 2	38.70	5.49	5.18	1.80
Aisle 3	42.66	4.95	4.87	1.45
Aisle 4	38.37	6.84	2.50	1.82
Aisle 5	38.35	3.86	11.43	1.04
Aisle 6	41.66	4.87	10.45	1.11
Aisle 7	44.18	3.88	13.14	0.83
Aisle 8	41.17	4.24	6.67	1.05
Aisle 9	41.73	4.35	3.22	3.40
Aisle 10	44.54	3.51	12.55	0.95
Mean	41.70	4.47	5.60	1.08

5 ACKNOWLEDGMENTS

A special acknowledgment to Sonae MCH, Portugal for providing the environment for data collection. The authors acknowledge the financial support obtained from the European Regional Development Fund (ERDF) through COMPETE - Programa Operacional Factores de Competitividade (POFC), for the project ShopView (project number 30335).

6 REFERENCES

[Kop09] J. Kopf, D. Lischinski, O. Deussen, D. Cohen-Or, and M. Cohen, "Locally adapted projections to reduce panorama distortions," in Computer Graphics Forum, 2009, vol. 28, pp.1083-1089.

[Ang10] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google Street View: Capturing the World at Street Level," Computer, vol. 43, no. 6, pp. 32-38, Jun. 2010.

[Kop10] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street Slide: Browsing Street Level Imagery," in ACM SIGGRAPH 2010 Papers, New York, USA, 2010, pp. 96:1-96:8.

[Aga06] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski, "Photographing long scenes with multi-viewpoint panoramas," in ACM Transactions on Graphics (TOG), 2006, vol. 25, pp. 853-861.

[Bro07] M. Brown, e D. G. Lowe. "Automatic Panoramic Image Stitching Using Invariant Features," International Journal of Computer Vision, vol. 74, no. 1, pp. 59-73, Aug. 2007.

[OpenCV] OpenCV documentation site: `http://docs.opencv.org/index.html`

[Low04] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", International journal of computer vision, vol. 60, no. 2, pp. 91-110, 2004.

[Wan08] Y. Wan and Z. Miao, "Automatic panorama image mosaic and ghost eliminating," in 2008 IEEE International Conference on Multimedia and Expo, 2008, pp. 945-948.

[Zis04] R. Hartley and O. Zisserman, "Multiple View Geometry in Computer Vision", second ed. Cambridge Univ. Press, 2004

[Dev95a] F. Devernay and O. D. Faugeras, "Automatic calibration and removal of distortion from scenes of structured environments," in SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation, 1995, pp. 62-72.

[Dev01b] F. Devernay and O. Faugeras, "Straight lines have to be straight,"Machine vision and applications, vol. 13, no. 1, pp. 14-24, 2001.

[Dud72] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15,1972, pp. 11-15.

[Per10] A. Pernek and L. Hajder, "Perspective Reconstruction and Camera Auto-Calibration as Rectangular Polynomial Eigenvalue Problem," in 2010 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 49-52.

[Mal07] E. Malis, M. Vargas, and others, Deeper understanding of the homography decomposition for vision-based control, 2007.

[Rub11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, pp. 2564-2571.

[Fis81] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Commun. ACM, vol. 24, no. 6, pp. 381-395, Jun. 1981.

[Sze11] R. Szeliski, Image stitching, in Computer Vision, Springer, 2011, pp. 375-408.

[Spi12] A. Spizhevoy and V. Eruhimov, "Problem of auto-calibration in image mosaicing," presented at the International Conference on Computer Graphics and Vision, GraphiCon, Conference Proceedings, 2012, pp. 27-32.

[Zhe11] E. Zheng, R. Raguram, P. Fite-Georgel, and J.-M. Frahm, "Efficient Generation of Multiperspective Panoramas," in 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 2011, pp. 86-92.

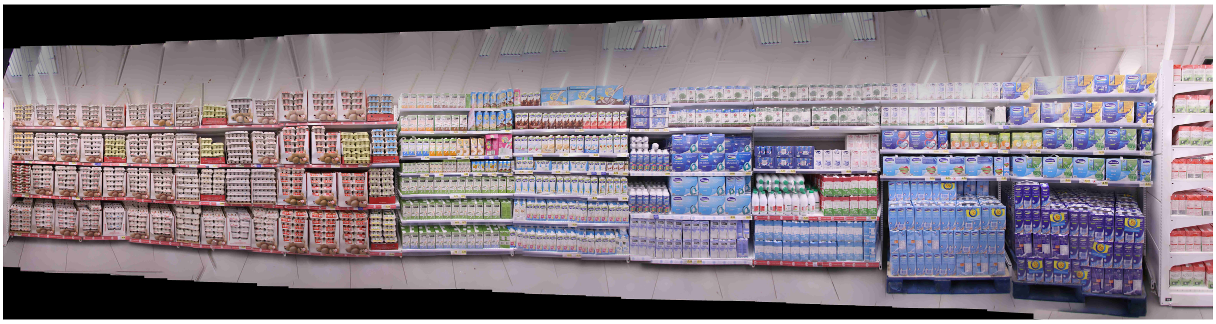


Figure 5: Panorama create with method from [Bro07] with uncorrected perspective effect.

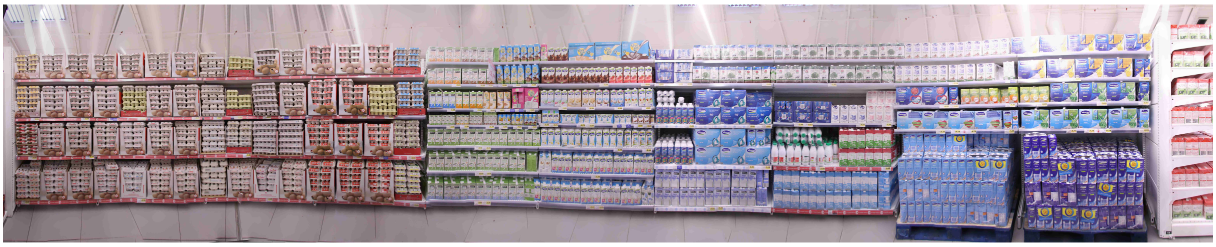


Figure 6: Panorama create with proposed method of the same aisle in Figure 5, with correction.

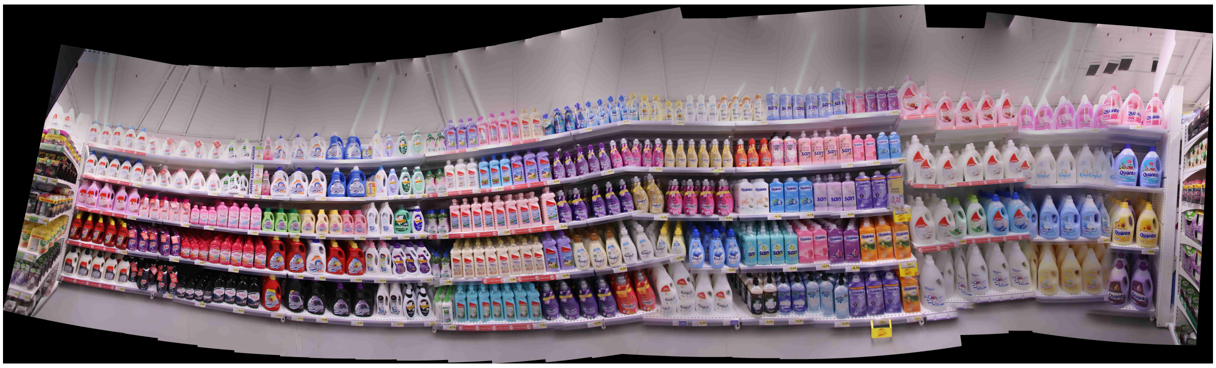


Figure 7: Panorama create with method from [Bro07] with missing products.

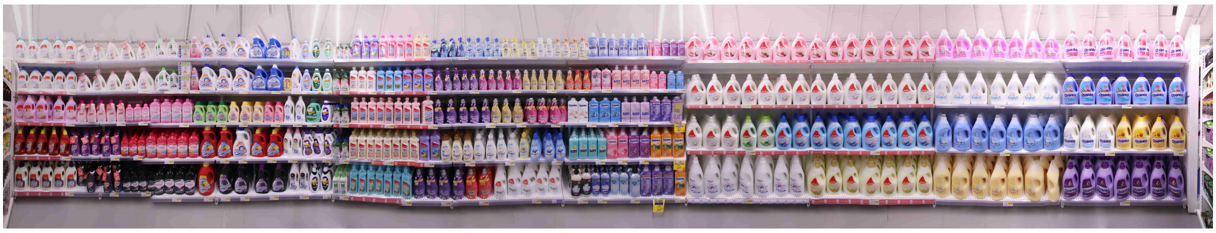


Figure 8: Panorama create with proposed method of the same aisle in Figure 7, with correction and right number of product items.

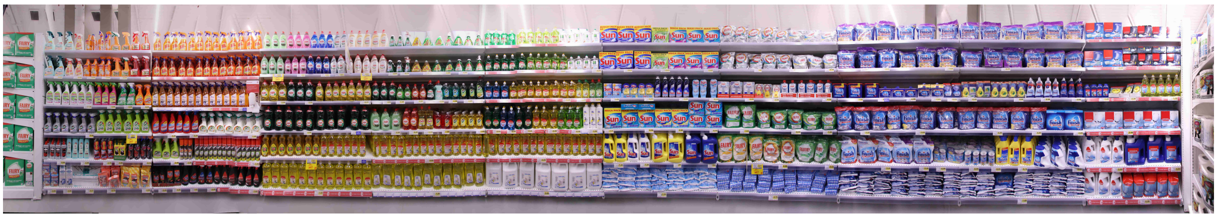


Figure 9: Another example of panorama create with proposed method with correction.

News Patterns: how press interacts with social networks

Guilherme Bertini Boettcher
Instituto de Informática - UFRGS
Caixa Postal 15064
Brazil 91501-970, Porto Alegre, RS
maiona@gmail.com

João Comba
Instituto de Informática - UFRGS
Caixa Postal 15064
Brazil 91501-970, Porto Alegre, RS
comba@inf.ufrgs.br

Carla Dal Sasso Freitas
Instituto de Informática - UFRGS
Caixa Postal 15064
Brazil 91501-970, Porto Alegre, RS
carla@inf.ufrgs.br

ABSTRACT

Social media has played a big part in the adaptation process for newspapers and magazines, but innovating while going through a recession has led to a hasty evolution and automated processes for very different media. While existing social media studies and state of the art visual solutions are available for analyzing social media content and users' behaviors, no other method is optimized for finding patterns from a popularity standpoint in the specialized realm of news channels. In this paper, we propose the usage of a combination of different visualization techniques that co-relate the profile's and its reading community activities with the resulting popularity. For the period of three months, we gathered Twitter posts, the number of followers and trending topics from worldwide press profiles. We used this dataset as the seed for our bar charts, tag clouds and bubble charts to allow for multiple source comparison, so that not only the user is able to understand their own community but also the success and pitfalls faced by the competition in the same medium. We validate our analysis by interviewing a group of journalists from different established newspapers. Through interacting with our system, it was possible to reveal hidden patterns in the massive dataset of messages and comments worldwide enabling the user to have unique insight into their community's behaviors and preferences.

Keywords

information visualization; social media; temporal patterns; diffusion patterns.

1 INTRODUCTION

In the last few years, we have witnessed a dramatic change in the way newspapers and magazines communicate, as well as in the time events take to spread around the world through the Internet. The decline in sales of print media has forced the press to adapt it's business to a more current media.

Facebook and Twitter, created as a social network and a microblogging service respectively, are popular social media venues that are recognized as relevant broadcasting and influence tools. Nowadays, they are commonly used by the press to generate interest for their published material and broadcast news. However, the adaptation from an established media to a new technology was hasty and for a variety of reasons, most newspapers simply utilize automatic publications to share their content on social media, with little to no research on their readers usage of such media.

The problem of identifying behavioral patterns in social media from a popularity standpoint is applicable in varied fields other than news. It can bring interesting insight for the fields of politics, entertainment or any other popularity driven networks that can benefit from finding key behavioral patterns in a social media.

In our research, we found solutions for parts of the problem optimized for a different general group of users. Among them, the works by Guodao et al. [SWL$^+$14], Yafeng et al. [LKT$^+$14], Yingcai et al. [WLY$^+$14] and Zhao et al. [ZGWZ14] [ZCW$^+$14] are the closest to a full solution. We were not able to find a complete solution for the problem focused on the press community from a popularity standpoint.

In this paper, we propose the usage of a combination of different visualization techniques that co-relate the profile's and its reading community activities with the resulting popularity.

We organize our text as follows: in session 1 we introduce the problem and present the applicable scenarios; in session 2, we discuss the recent and state of the art techniques, their advantages and shortcomings; in session 3, we present our proposed solution in detail; in session 4, we analyze the experimental results obtained in three different use cases and expert reports; in session 5, we provide a review of the technique, its advantages and shortcomings and final comments; and in sessions 6 and 7, we present our acknowledgments and references.

2 RECENT SOLUTIONS

There are several works dealing with social networks analysis and visualization. As mentioned before, most are relatively simple tools to obtain statistics about users and the overall network. Literature closely related to our work can be roughly divided in two categories: analysis of information diffusion processes and visualization of diffusion patterns.

2.1 Analysis of information diffusion processes

Social networks have been studied for years [WF94], but online social networks, blogs and microblogs introduced challenges in the investigation of how people communicate using these media. The analysis of information diffusion in such networks involve measuring quantitative characteristics [KLPM10, YW10], finding relations between structure and dynamics [LG10, YC10a], predicting characteristics of the diffusion process [YC10b] and trends detection approaches [MK10, CTB$^+$12].

In addition to processing the collected data, most of these works rely on showing static plots to display the values of the metrics they are concerned about. As for trend detection, the tools must process information in real-time. For example, Twittermonitor [MK10] produces a webpage reporting recent trends in real time and provides an interface for users to rank trends according their own criteria.

2.2 Visualization of information diffusion patterns

Besides the many tools that provide graphical ways for monitoring social media activity, there are recent works that propose the visualization of information diffusion patterns [KLPM10, YC10a, CTB$^+$12, SWL$^+$14, LKT$^+$14, WLY$^+$14, ZGWZ14, ZCW$^+$14]. Among them, works of Sun et al. [SWL$^+$14], Lu et al. [LKT$^+$14], Wu et al. [WLY$^+$14] and Zhao et al. [ZGWZ14, ZCW$^+$14] are the closest ones to a full solution, and we will restrain ourselves to briefly describe them.

Sun et al. [SWL$^+$14] and Wu et al. [WLY$^+$14] aim at analyzing topics coopetition in social media (most notably, Twitter) and answer who exerts the greatest influence on a highly cooperative topic that used to be a competitive topic, what are the similarities and differences in the roles of groups of issue publics and how often they divert attention to other topics. Their tool summarizes dynamic topic competition and compares topic leaders to topics by utilizing the Theme River technique described by Havre et al. [HHN00], however is focused on the patterns of cooperation versus competition of different themes and does not contemplate the analysis of individual profiles.

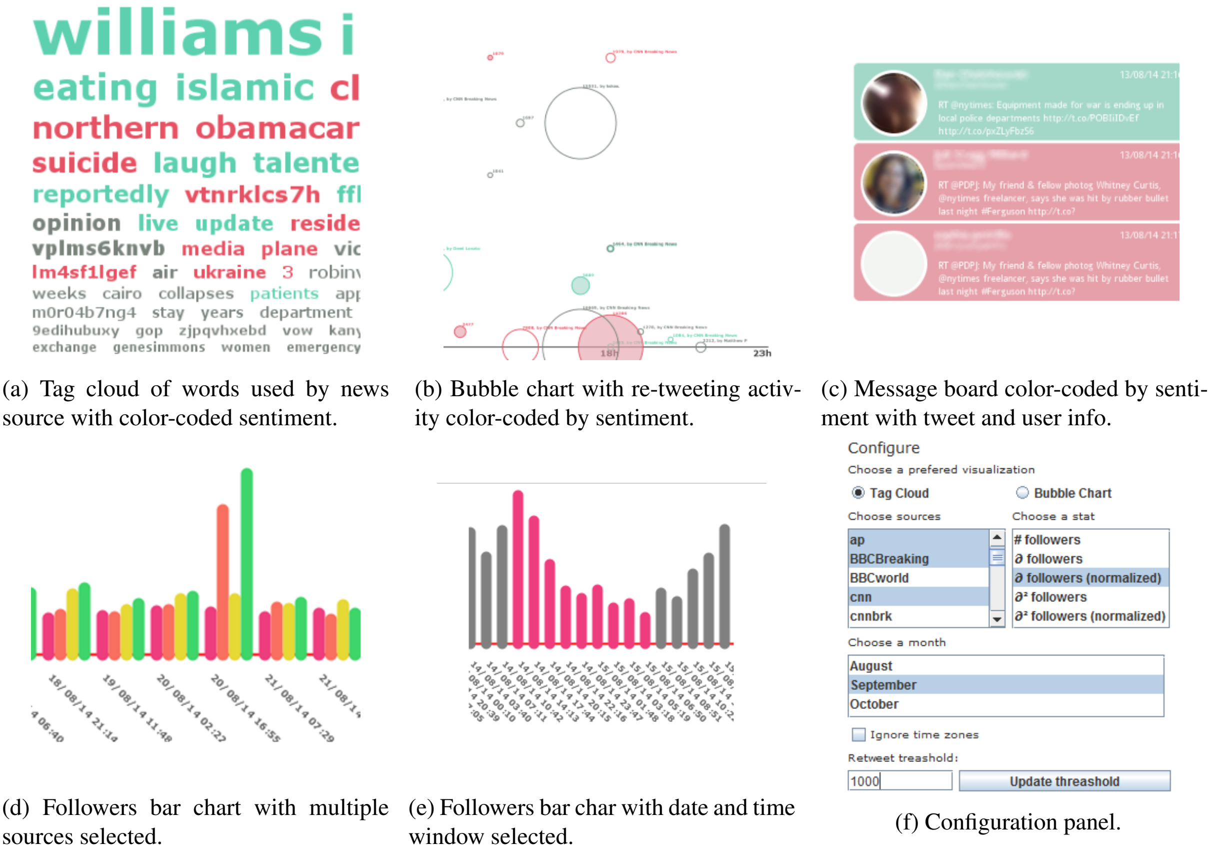
Lu et al. [LKT$^+$14] describes a framework for predictive models using social media (IMDB.com, Twitter and Youtube) in an attempt to create a tool that enables non-domain experts to be competitive with experts in a given area. The tool combines line, bar, bubble and candle charts, parallel coordinates and a tag cloud with sentiment analysis as means for the user to explore the information available in the mentioned media and choose from calculated metrics to predict the popularity of movies in their opening weekends. Even though they provide a myriad of visualization techniques, their solution does not enable the user to identify patterns for the increase or decline of popularity of the subject.

Zhao et al. [ZGWZ14] describes a comprehensive tool for analysis of behavior emotion and ultimately mood of a given person in social media (most notably Twitter) over time. They provide a multi-dimensional emotion analysis tool with the ability of extracting emotional episodes and infer longer-lasting moods through an enhanced implementation of Havre et al. [HHN00] and rich interaction, however this analysis is not applicable to press profiles, since they use their microblogs to increase visibility of their online content, instead of commenting on their own day-to-day activities and is not scalable or reliable.

Finally, Zhao et al. [ZCW$^+$14] describes a system for the detection, exploration and interpretation of anomalous conversational threads in Twitter. This solution is applicable to the news environment, as we could consider an abnormal increase or decline in popularity of a given profile as an anomaly and apply their algorithms to further explore the available data. The problem lies in the nature of the anomalies which is the lack of a clear definition. A thread may be considered abnormal when it disseminates a message differently from the patterns of other threads in a similar topic. This is not necessarily true, especially when we take a channels popularity into account. Common subjects are covered by different profiles in unique ways that would be considered anomalous by their solution.

3 PROPOSED SOLUTION

We proposed the use of a combination of bar chart, bubble chart, tag clouds and message boards coupled with responsive interaction among the combined visualization techniques to bridge the gap left by other studies in the same area.


williams i
eating islamic cl
northern obamacar
suicide laugh talente
reportedly vtnrklcs7h ffl
opinion live update reside
vplms6knvb media plane vic
lm4sf1lgef air ukraine 3 robin
weeks cairo collapses patients ap
m0r04b7ng4 stay years department
9edihubuxy gop zjpqvhxebd vow kan
exchange genesimmons women emergency
18h
23h
18/08/14 21:14
19/08/14 11:49
20/08/14 02:22
20/08/14 16:55
21/08/14 07:29

13/08/14 21:1

RT @nytimes: Equipment made for war is ending up in local police departments http://t.co/POBIiIDvEf http://t.co/pxZLyFbzS6

13/08/14 21:1

RT @PDPJ: My friend & fellow photog Whitney Curtis, @nytimes freelancer, says she was hit by rubber bullet last night #Ferguson http://t.co?

13/08/14 21:1

RT @PDPJ: My friend & fellow photog Whitney Curtis, @nytimes freelancer, says she was hit by rubber bullet last night #Ferguson http://t.co?

Configure

Choose a prefered visualization

● Tag Cloud ○ Bubble Chart

Choose sources: ap, BBCBreaking, BBCworld, cnn, cnnbrk (selected: ap, BBCBreaking, cnn)

Choose a stat: # followers, ∂ followers, ∂ followers (normalized), ∂² followers, ∂² followers (normalized) (selected: ∂ followers (normalized))

Choose a month: August, September, October (selected: September)

□ Ignore time zones

Retweet treashold: 1000

Update threashold

(a) Tag cloud of words used by news source with color-coded sentiment.

(b) Bubble chart with re-tweeting activity color-coded by sentiment.

(c) Message board color-coded by sentiment with tweet and user info.

(d) Followers bar chart with multiple sources selected.

(e) Followers bar char with date and time window selected.

(f) Configuration panel.

Figure 1: Different modules of the News Patterns

We combined analytical and statistical information from original posts by news profiles and readers alike, channel popularity information and trending topics. We included configuration options that allow the user to filter out any activity that falls outside of the subject of interest and concentrate on the actions that closely relate to the popularity spikes and valleys found by the user. We took the user's expertise into account when defining which are the abnormal activities in the available dataset, since automatic search for such patterns falls into the nature of true anomalies for which there is no clear definition.

In order to gain insight into the daily work of the target users for this system and how it could benefit them in obtaining better results from their efforts in social media, we conducted interviews with journalists from different major newspapers. Based on the gathered information, we tailored our solution to answer the key questions posed by the group of journalists:

1. Why do readers stop following a profile?
2. Does the time of the post co-relate with the number of retweets?
3. How does the profile relate with it's network?
4. Is there correlation with posting subjects and the amount of followers/popularity of a profile?

After creating a working prototype, we shared our proposed solution and monitored their usage. We mention their findings and analyze their reports regarding the tool in the results chapter.

3.1 Data gathering

Data was gathered from Twitter over the period of 3 months with the use of the Firehose API, by filtering the messages by original author, retweeted status author or user mention to match the list of 19 news source Twitter IDs spread worldwide. The total numbers amount to 15 million original tweets from August 2014 to October 2014, averaging 5 million publications per month and over 10.000 posts per source per day. Each status update object is stored raw in a JSON format on plain files. Complementing this dataset, further data was gathered from each source, containing the number of followers every 30 minutes for the same time period, resulting in an average of over 1000 snapshots per source per month. Finally, a third and separate dataset gathered the trending topics from the 15 available locations in Brazil every 30 minutes, resulting in 495 unique trending topics over the three month period (an average of 165 unique trending topics per month).

3.2 Tag Cloud

The tag cloud shows the most recurrent words and the sentiment of the messages relating to them for the selected profiles during the time period defined by the user.

The visualization is implemented as displayed in figure 1a, with either the words from selected profiles or with a localized list of trending topics. Like most implementations of this technique, occurrence of the word is mapped to it's font size, so that the most commonly used words or tags will be largest as well. Most users will be using this visualization to understand which subjects were of interest in the selected window of time, so to augment the importance of each term, the most common words are also listed first in the visualization.

The sentiment attached to each word will be calculated from the messages that contain them. We used the SentiWordNet 3.0 lexical resource for opinion mining, which assigns to each synset of WordNet three sentiment scores: positivity, negativity and objectivity. After we calculated the accumulated sentiment of all messages, we color-coded the word in a green hue if the result is positive and above a configurable threshold, red if the result is negative and below a configurable threshold or gray otherwise. Restraining the seeds to account only for the profile's original posts will create a picture of their sentiment regarding a specific topic.

3.3 Bubble Chart

The bubble chart displays the tweeting activity as shown in figure 1b for the selected source and time period. The X axis maps the hour of the day when the message was posted and the Y axis to map the popularity the selected profile(s) had during the analyzed period, ranging from the smallest to the largest amount of followers.

The size of the bubble maps the number of re-tweets for the original message in the moment of the post. As part of the nature of the fire-hose API, we do not have guaranteed access to all the messages being posted at any time, but we can compare the number of re-tweets of the original post to get an idea of how popular each message became over time.

The bubbles are color-coded with the information as discussed in 3.2, with the added information of agent mapped to the fill of the bubble. Readers can comment or simply re-tweet messages (I) directly from the source, but the same activity may come from (II) a separate profile that mentioned, commented or re-tweeted the news channel. In figure 1b, scenario (I) is represented by filled bubbles and (II) is mapped as hollow circles. In both cases, we include the raw number of re-tweets and in case (II), we displayed the screen name of the profile that generated the activity.

3.4 Message Board

Figure 1c shows original message information that belongs to the selected sources and date and time being analyzed. As in the other modules, sentiment is color coded as the background for each message. We included the original text, the date and time of the posting, the author's name, screen name and profile picture. This module allows us to get specific information in it's lowest granularity and rawest form, which can help the user identify the context from which the tag cloud and bubble charts were derived.

Clicking any message filters out every word or bubble from unrelated posts, enabling in-depth analysis of any conversation of interest.

3.5 Profile followers bar chart

The followers bar chart displays the patterns of popularity of the selected profiles according to the chosen metric over time as shown in figure 1d.

This module is very helpful in determining the relationship each news source has with their followers in terms of popularity. It allows the user to discover patterns in the growth of popularity of any channel and for the comparison between channels in a leveled playing field. This graph is a good starting point to discover points of interesting activity during an extended period of time. The user can select the appropriate window to restrain the information to the activity as shown in figure 1e in order to research the window of interest and the remaining panels will filter out any information not comprised in that period.

This implementation is fundamental to create a popularity-based solution, not found in any other work of similar proposals.

3.6 Configuration

The configuration panel is organized as in figure 1f, where users can choose the switch from each visualization type (Tag cloud and Bubble chart) and choose from the list of available profiles the ones they are interested in analyzing at each time. Choosing from different metrics of behavior, i.e. total number of followers, delta from each interval to the next, normalized delta from each interval to the next, delta squared and normalized delta squared of each interval. These two selections are necessary to display the bar chart of followers, shown in figure 1d.

To optimize the search performance, we separated the dataset into each month (August, September and October), so the user can easily switch between time windows of interest. The user also has the option of ignoring or taking the time zone into account while calculating the statistics. This separation is key to improve scalability.

The last item in the configuration panel is the re-tweet count threshold for the bubble chart. Only posts that surpass the minimum amount specified in that field will be displayed. This feature is particularly helpful when studying popular channels, such as CNN and the NY Times.

4 EXPERIMENTAL RESULTS

In order to determine the efficacy of our proposed solution, we invited the same corpus that was present during the initial phase of the project to test the tool and use their on expert knowledge, explore the existing data to find answers for the four main questions presented the introduction section. We utilize key use cases to illustrate their reported findings and our analysis.

4.1 Use case 1

Comparison between different sources is not particularly key to understand the patterns of a profile's following base, but it can be very helpful in determining what other news sources have been doing and how that affected their popularity.

According to the journalists we interviewed, most of their activity in Twitter is automated, which means that information about their following crowd is not regarded while posting. When a new piece is included in the print version, it is automatically added in the channel's online publication after the editors finalize the next morning's issue. After the issue goes live, automatic processes create the messages and posts them automatically at the same time.

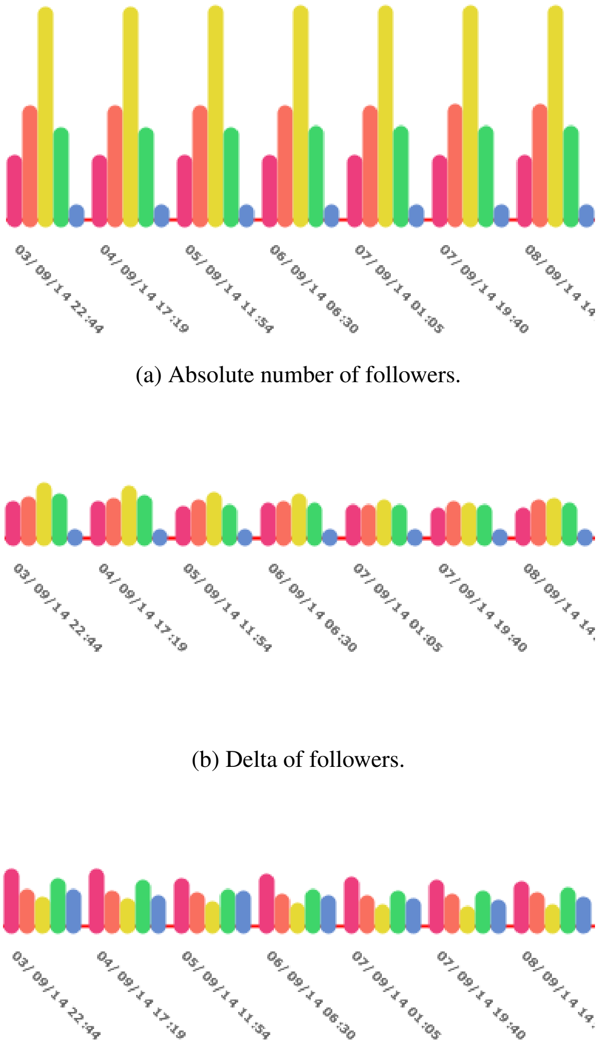
While worldwide news sources such as BBC and The Economist have a large follower count, the user is still able to compare them to local sources with a more limited reach, in order to understand if the community behavior follows a global trend or if their are driven by different motives.

As a filtering activity, the user selects the appropriate window to restrain the information to the activity as shown in figure 1e in order to research the pattern revealed and select either the tag cloud or bubble chart visualization to better understand which were the causes of that behavior.

Three different views of the followers trends are shown in figure 2. There are five selected profiles (Associated Press, BBC, CNN, The Economist and The Sun) based on three different metrics (Absolute number of followers, delta and normalized delta).

It is clear that CNN has a commanding lead in popularity, while The Sun has a very limited reach on Twitter, based on figure 2a. It is also clear that the difference in popularity is sustained over time by each of the sources.

We can see in figure 2b that the number of followers increases by a similar amount for each of the profiles, The Sun being the sole exception. However, when we analyze figure 2c, it becomes clear that the effect of growth is actually greater for the Associated Press profile. This analysis means that even though it may take a long time, the Associated Press and The Economist are diminishing the gap of popularity between them and BBC.



(a) Absolute number of followers.

(b) Delta of followers.

(c) Normalized delta of followers.

Figure 2: Followers trends for Associated Press, BBC, CNN, The Economist and The Sun.

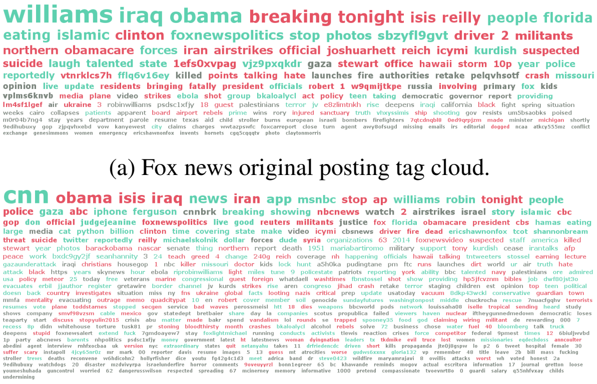
4.2 Use case 2

To understand the relationship between a given profile and it's readers, we break their interaction into four aspects:

- Most common words used
- Time of original posts
- Time of retweets, comments and other related activity by the community

- Sentiments associated with each of the above

During the analysis of the tool, we selected Fox News as the primary source to be analyzed and noticed the same pattern of popularity repeating itself over time, so we selected that window of dates, from August 9th to 11th. Figure 3a shows the most used words for Fox News in that time.



(a) Fox news original posting tag cloud.

(b) Fox News readers comments tag cloud.

Figure 3: Comparison of original content provider and the reader community activity.

It is clear that the subjects of interest during this period for the profile were president Obama, Iraq, Robin Williams, ISIS and O'Reilly, which is not surprising since during that time, US jet fighters launched a strike on ISIS militants and Robin Williams came to pass.

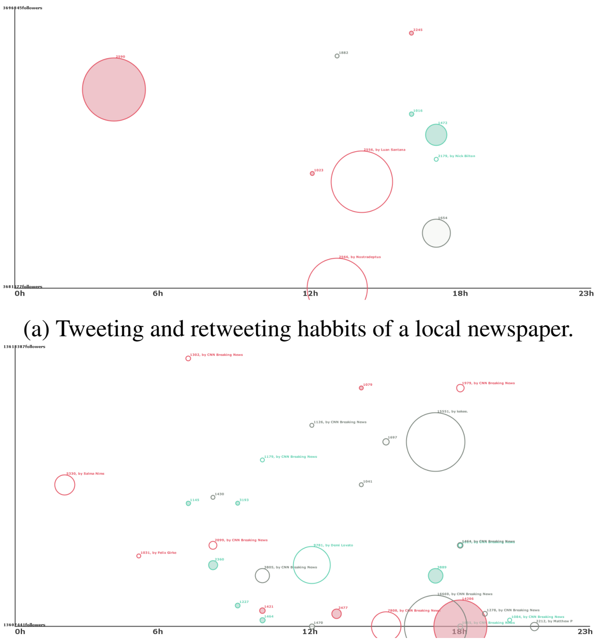
What is revealing is that even though Fox News is considered to have a republican bias, according to the analysis, they choose positive words when discussing president Obama, Iraq and the Islamics, while using negative words when mentioning their on-camera talent O'Reilly. Our users were expecting the opposite trend to be in place. Only the specific issues of "Obamacare" and Sen. Clinton are mentioned in a negative post, in confirmation to the journalists expectations.

By filtering the seeds to include the reading community's posts alone, the user is able understand their opinions in the realm of the channels messages. Figure 3b shows the Fox News' community mentions the president in a contrasting negative fashion, while confirming the position on issues like ISIS. Another interesting contrast is the related profile @foxnewspolitics, which is mentioned positively by the main account but negatively by the community. This contrast poses a very interesting insight: even though Fox News may at times be favorable to the democratic issues, they community still mentions their content in a negative scenario.

In use case 2, we will describe further cases that relate to the time patterns found for other channels.

4.3 Use case 3

The bubble chart technique allows us to understand how time and popularity influence the activity in any given message as well as to identify if when there are any key readers generating the interest instead of the original poster.



(a) Tweeting and retweeting habits of a local newspaper.

(b) Tweeting and retweeting habbist of CNN.

Figure 4: Comparison of original content provider and the reader community activity.

While the popularity of local and global news sources vary greatly, it is possible to identify similar trends in both communities: most of the activity takes place between noon and dusk. However, figure 4 shows how the local newspaper relies on separate influencer's retweeting or commenting of their post, most of the activity from the global news channel community is channeled through profiles from outside the institution. In contrast, the global source displays activity across the entire Y index, indicating that current popularity is not particularly key for generating large amounts of activity.

Figure 4a, is an interesting case for the relationship of popularity and activity. Contrary to our initial estimates, activity from the community is inversely related to the popularity of the channel. After finding this pattern in our database in repeated occasions, we investigated the relationship of the source with the community via the message board and discovered that this particular source is heavily dependent on key influencers to raise their popularity, even though this was not the case for actual activity related to their posts. Figure 4a also displays two significant influencers found in this local newspaper's community that will consistently significantly increase activity around topics of interest, generating the greatest re-tweet counts of this particular source throughout the entire three month available period. This specific discovery made by one journalist was deemed important to determine who are the key people that the publication can involve in order to increase its popularity.

4.4 Journalist reports

We organized a dedicated guided testing session of the tool with each of the journalists, which was comprised of a 20 minute tutorial of the visualizations and interactions followed by a 40 minute assisted free play and a final questionnaire section to determine the tools efficacy. Even though the majority of them found that performance was an issue, 75% reported that the interaction was very intuitive and provided a clear way to investigate any area of interest.

Most of the patterns found by the senior journalists were expected, based on their gathered knowledge over the years of work, however the junior journalists were more intrigued by their findings. This points to the direction that the tool may be more beneficial to unexperienced professionals as streamlined way to understand their public.

We improve on existing information diffusion pattern seeking studies by enabling (1) the discovery behavioral patterns that influence and explain increase and decline of popularity of any specific channel, (2) the discovery of key third party influencers that actively shape the community's interests and (3) the comparison of similar and distinct sources' communities in a variety of ways.

5 DISCUSSION AND FINAL COMMENTS

Our solution utilizes known visualization techniques in innovative manners, aggregating information such as sentiment and popularity to give a unique view of the behavior of the community surrounding the news source profile and their relationship with each other.

Through the use of the proposed solution, it was possible to reveal hidden patterns and gain insightful knowledge of the reading community which address real needs from the industry. The system improves the existing pool of solutions by using very clear parameters and established techniques to provide solutions to otherwise unanswered questions. While there are some gaps between the proposed motivations and our solution, based on the experts reports we understand that it shows considerable promise and believe that by enhancing the existing interaction, we will able to provide an important tool that can fine tune the way the press interacts with social networks.

The system produces convincing results for most scenarios but is not without its limitations. As mentioned in 4.4, performance remains an issue especially for live data due to the heavy calculations needed to appropriately enable the interactions between panels. Complexity analysis on our heaviest algorithm tend to $O(n^2+k)$ which, like Zhao et al. [ZCW$^+$14] poses issues with scalability. The added functionalities make the system more powerful, but also more complex. Depending on the unique patterns of a given community's activity, performance may also be affected negatively.

6 ACKNOWLEDGMENTS

We would like to thank the journalists that helped us during the development of this study. We also acknowledge the financial support from the Brazilian funding agencies CNPq, CAPES and FAPERGS.

7 REFERENCES

[CTB$^+$12] Junghoon Chae, D. Thom, H. Bosch, Yun Jang, R. Maciejewski, D.S. Ebert, and T. Ertl, *Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition*, Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on, 2012, pp. 143–152.

[HHN00] S. Havre, B. Hetzler, and L. Nowell, *Themeriver: visualizing theme changes over time*, Information Visualization, 2000. InfoVis 2000. IEEE Symposium on, 2000, pp. 115–123.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon, *What is twitter, a social network or a news media?*, WWW Proceedings, 2010, pp. 591–600.

[LG10] K. Lerman and R. Ghosh, *Information contagion: An empirical study of the spread of news on digg and twitter social networks*, ICWSM, 2010.

[LKT$^+$14] Yafeng Lu, R. Kruger, D. Thom, Feng Wang, S. Koch, T. Ertl, and R. Maciejewski, *Integrating predictive analytics and social media*, Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on, Oct 2014, pp. 193–202.

[MK10] Michael Mathioudakis and Nick Koudas, *Twittermonitor: trend detection over the twitter stream*, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (New York, NY, USA), SIGMOD '10, ACM, 2010, pp. 1155–1158.

[SWL$^+$14] Guodao Sun, Yingcai Wu, Shixia Liu, Tai-Quan Peng, J.J.H. Zhu, and Ronghua Liang, *Evoriver: Visual analysis of topic coopetition on social media*, Visualization and Computer Graphics, IEEE Transactions on **20** (2014), no. 12, 1753–1762.

[WF94] Stanley Wasserman and Katherine Faust, *Social network analysis: Methods and applications*, vol. 8, Cambridge university press, 1994.

[WLY+14] Yingcai Wu, Shixia Liu, Kai Yan, Mengchen Liu, and Fangzhao Wu, *Opinionflow: Visual analysis of opinion diffusion on social media*, Visualization and Computer Graphics, IEEE Transactions on **20** (2014), no. 12, 1763–1772.

[YC10a] Jiang Yang and Scott Counts, *Comparing information diffusion structure in weblogs and microblogs*, ICWSM'10, 2010.

[YC10b] ______, *Predicting the speed, scale, and range of information diffusion in twitter*, ICWSM, 2010.

[YW10] Shaozhi Ye and S. Felix Wu, *Measuring message propagation and social influence on twitter.com*, Proceedings of the Second international conference on Social informatics (Berlin, Heidelberg), SocInfo'10, Springer-Verlag, 2010, pp. 216–231.

[ZCW+14] Jian Zhao, Nan Cao, Zhen Wen, Yale Song, Yu-Ru Lin, and C. Collins, *Fluxflow: Visual analysis of anomalous information spreading on social media*, Visualization and Computer Graphics, IEEE Transactions on **20** (2014), no. 12, 1773–1782.

[ZGWZ14] Jian Zhao, Liang Gou, Fei Wang, and M. Zhou, *Pearl: An interactive visual analytic tool for understanding personal emotion style derived from social media*, Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on, Oct 2014, pp. 203–212.

A Geodesic Based Approach for an Accurate and Invariant 3D Surfaces Representation

Majdi Jribi
CRISTAL Laboratory,
GRIFT research group
ENSI,La Manouba
University
2010, La manouba,
Tunisia
majdi.jribi@ensi.rnu.tn

Faouzi Ghorbel
CRISTAL Laboratory,
GRIFT research group
ENSI,La Manouba
University
2010, La manouba,
Tunisia
faouzi.ghorbel@ensi.rnu.tn

ABSTRACT

In this paper, we propose a novel 3D invariant surface representation under the 3D motion group $M(3)$. It is obtained by combining two main representations: the three-polar representation and the one defined by the radial line curves from a starting point. The retained invariant points correspond to the geometrical locations of the intersection between the two last representations. The approximation of the novel surfaces description method on the 3D discrete meshes is studied. Its accuracy for the 3D faces description and retrieval is evaluated in the mean of the Hausdorff shape distance.

Keywords

Three-polar, geodesic potential, superposition, level set, radial line, shape representation, Hausdorff, invariant, approximation, face.

1 INTRODUCTION

Actually, it is recognized that the 3D data have allowed to cross several planar images (2D) problems such as the illumination and the pose. Many domains benefit from the use of the three dimensional data like the medical one, the security and the heritage conservation.

Despite the amazing development of the 3D data scanning tools and the related technologies, the description and the analysis of 3D surfaces remain a difficult task. This comes from the lack of a natural parametrization of a 3D surface. In fact, the obtained data depend largely on the position of the scanning tool relatively to the 3D object.

In practice, the conventional representation of a 3D surface is the discrete triangulated mesh. The points cloud composing this conventional representation is not organized or partially organized. This fact made the comparison procedure between different shapes more and more difficult. Therefore, the extraction of an efficient description of the 3D surfaces has become more than necessary. In the literature, the 3D surface description methods can be grouped into four major classes: the two dimensional views, the graphs based approaches, the transformations based ones and those obtained by statistical features.

The two dimensional views based methods suppose that a 3D object corresponds to a set of 2D images. They are obtained by the projection of the 3D object from canonical points of views. The task of describing a 3D object is transformed, therefore, to a description of planar images. Many description methods can be applied on the obtained 2D images as Fourier descriptors [Vra04] and Zernike moments [Che03].

The approaches based on the graphs consist on the extraction of graphs from a 3D object. The comparison between different shapes is, therefore, reduced to a comparison between their corresponding graphs. One of the most used descriptors is the Reeb graph [Tun05]. This graph is invariant under the rotations and the translations transformations. It is also robust under the connectivity changes and the mesh simplifications. These characterizations made this last method an accurate tool for 3D surfaces description. The skeletal method [Sun03] is also known as a perfect tool for 3D surfaces description based on the graphs.

The transform based methods need first of all the conversion of the surface onto 3D voxels or a spherical grid. Many specific transformations are, then, applied to extract efficient description of the surface. The most famous methods are the 3D Radon [Dar04], 3D Fourier [Bur92] and the rotation-invariant spherical harmonics [Kaz03]. These methods of transformations are characterized by their invariance under the rotations, the translations and the scale factors. Their accuracy for the recognition and the retrieval is proved. Other works use the angular radial transform [Ric05], the spherical wavelet descriptors [Lag06] and the uniformization [Khe08] as methods of description based transformations.

The statistical features based methods consist on the extraction of numerical attributes of a 3D object which can be local or global. Many past works adopted this approach for the extraction of an efficient 3D surfaces description. We mention the pioneer work of Faugeras et al. [Fau86] based on the high curvature area determination. A 3D surface is considered as a set of points that correspond to the geometrical locations of the high curvatures values. These points are invariant under the rotations and the translations transformations. Bannour et al. [Ban00] proposed a novel pseudo-reparametrisation of 3D surfaces by the extraction of iso-curvature features. In this method of description, a 3D surface is characterized by a curves network determined by iso-curvatures computation. The surface here is not described by the points corresponding to the extremal curvatures values but by the ones corresponding to the levels set of the curvatures values. Other kind of methods use the geodesic approach which is more stable under the numerical computation errors than the ones based on the curvatures determination. Many works use the local coordinate system computed around one reference point of the surface [Sam06, Sri08, Gad12] qualified by the unipolar representation (one reference point). It consists on the set of points corresponding to the levels of the geodesic potential generated from one reference point. In recent works, Jribi et al. [Jri13, Jri14] proposed a novel representation that they qualified by the three-polar one. It is defined by the levels of the sum of the three geodesic potentials generated from three reference points. This representation has ensured a more stability under errors on the reference points positions than the unipolar one [Jri14].

We propose in this paper to extract accurate invariant points from the three-polar representation. In order to achieve this purpose, we combine the three-polar representation with the one defined by the radial line curves constructed from a starting point. The novel set of invariant points is obtained by the intersection between the two last representations. The steps of the construction of the novel representation from the 3D discrete meshes will be studied. Its accuracy for 3D faces description will be evaluated.

Thus, this paper will be structured as follows: In the second section, we will present the mathematical formulation the novel representation. The similarity metric to compare between different shapes with the invariant points cloud will be exposed in the third section. In the fourth section, the approximation of the novel representation on the 3D discrete meshes will be presented. We will show in the fifth section, its accuracy for the 3D faces recognition and retrieval.

2 CONSTRUCTION OF THE NOVEL REPRESENTATION

We present in this section the mathematical formulation of the novel representation. We suppose, here, that a surface is continuous. Thus, it corresponds to a two differential manifold denoted S. Let denote by r and q two points of S. We start by presenting some differential considerations. We designate by:

- $\gamma(r,q)$: The geodesic curve joining r and q. It corresponds to the curve on the surface S having the minimal distance between r and q.
- $\widetilde{\gamma}(r,q)$: The length of the geodesic curve joining r and q.
- $U_r : S \to R$: the function that computes for each point p of S the length of the geodesic curve joining p to r. $U_r(p) = \widetilde{\gamma}(r,p)$. It is called the geodesic potential generated from the reference point r of the surface S.
- $C_U^\lambda = \{p \in S; U(p) = \lambda\}$: The geodesic level curve of value λ. It corresponds to the set of points of S which have the same value λ of the geodesic potential U. U can be a geodesic potential generated from one reference point or the sum of several geodesic potentials.

Three main steps are realized in order to obtain the novel representation:

- Construction of the three polar representation.
- Extraction of the representation defined by the radial line curves from a starting point.
- The intersection between the two last differential representations.

The obtained invariant points of the novel representation correspond to the intersection between the three polar representation and the one composed by the radial line curves.

2.1 Brief recall of the three-polar representation construction

We present, here, a brief recall of the construction process of the three-polar representation [Jri14]. It is based on the superposition of the three geodesic potentials generated from three reference points of the surface. It consists on the set of points corresponding to the levels of the sum of the three geodesic potentials. Let $\{r_i, i = 1..3\}$ be three reference points of the surface and $\{U_{r_i}, i = 1..3\}$ their corresponding potentials functions. $U_3 = \sum_{i=1}^{3} U_{r_i}$ is the geodesic potential obtained by the sum of these three geodesic potentials. Then, the three-polar representation denoted by $T_3^K(S)$ can formulated as follows:

$$T_3^K(S) = \{p \in S; U_3(p) = U_3^* + \frac{k}{K}(\alpha_K - U_3^*), k = 0..K\} \quad (1)$$
$$= \{C_{U_3^*}^{\lambda_k}; \lambda_k = U_3^* + \tfrac{k}{K}(\alpha_K - U_3^*), k = 0..K\}.$$

Where K is the number of the levels of the three-polar representation, α_K is the maximum of the geodesic sum, $U_3^* = min\{U_3\}$ and the integer k designates the k^{th} level of the three polar representation.

We note that this representation is invariant under the $SO(3)$ rotation group and the displacement one. The obtained three-polar representation is composed by a collection of indexed level curves according their level values.

2.2 Construction of the radial line curves representation on a 3D surface

Let q be a point of the surface S. Let denote by P_q^0 a plane that contains the point q and that intersects the surface S on a curve (radial line curve) that we call the reference radial line curve and we denote by R_q^0. The choice of R_q^0 depends on the kind of the surface.

Let call by R_q^α the radial line curve making an angle α with the reference radial curve R_q^0. It is obtained by the intersection between the plane P_q^α and the surface S. P_q^α is the plane containing the point q and having the angle α with the reference plane P_q^0. Since this plane is not unique, we choose a clockwise direction. We repeat the process of radial line curves extraction with the same angular separation.

This representation that we denote by RL^K (Radial lines) is formulated as follows:

$$RL^K(S) = \{P_q^{k\alpha} \cap S; k = 0..K\}. \quad (2)$$

We obtain, therefore, an indexed collection of radial line curves on the surface.

2.3 The obtained invariant points of the novel representation

From the construction of the three-polar representation, we obtain an indexed level curves of the sum the three geodesic potentials generated from their corresponding three reference points.

A collection of radial line curves indexed by their angular values according to the reference radial line curve is also obtained by constructing the radial line curves from a starting point of the surface.

By computing the intersection between the level curves of the three-polar representation and the radial line curves representation, we obtain a set of invariant points indexed by both the value of the level curves of the three-polar representation and the angle of the radial line according to the reference one.

The novel representation defined by this selection of invariant points will be denoted by $3PRL$ (Three-Polar and Radial Lines).

$$3PRL(S) = RL^K(S) \cap T_3^M(S). \quad (3)$$

3 SIMILARITY METRIC

We propose to compare between the different shapes using their corresponding novel representations. We use the well known Hausdorff shape distance introduced by Ghorbel in [Gho98, Gho12]. Let consider the real plane R^2 and the unit sphere S^2 that represent the group G of all possible normalized parametrisations of surfaces. Since the space of surfaces pieces can be seen as a set of all 3D objects assumed diffeomorphic to G, this space is assimilated to a subspace of $L^2_{R^3}(G)$ formed by all square integrated maps from G to R^3. The direct product of the Euler rotations group $SO(3)$ by the group G , acts on such space $L^2_{R^3}(G)$ in the following sense:

$$SO(3) \times G \times L^2_{R^3}(G) \rightarrow L^2_{R^3}(G) \quad (4)$$
$$\{A, (u_0, v_0), S(u,v)\} \rightarrow AS(u + u_0, v + v_0).$$

The 3D Hausdorff shape distance Δ can be written for every S_1 and S_2 belonging to $L^2_{R^3}(G)$ and g_1 and g_2 to $SO(3)$ as follows:

$$\Delta(S_1, S_2) = max(\rho(S_1, S_2), \rho(S_2, S_1)). \quad (5)$$

Where:

$$\rho(S_1, S_2) = \sup_{g_1 \in SO(3)} \inf_{g_2 \in SO(3)} \| g_1 S_1 - g_2 S_2 \|_{L^2}. \quad (6)$$

$\| S \|_{L^2}$ denotes the norm of the functional space $L^2_{R^3}(G)$.

We consider after that, a normalized version of Δ so that its variations are confined to the interval [0,1]. This distance is obtained by using the well known Iterative Closest Point (ICP) algorithm [Bes92].

4 APPROXIMATION OF THE 3PRL REPRESENTATION ON THE MESHES OF 3D FACES

The description and the analysis of the shapes of 3D faces have achieved an increasing importance in the last few decades especially with the great development of 3D scanning tools. In all steps of the 3PRL representation construction, we supposed that the surface is a continuous two differential manifold. In the practice, the data obtained from the 3D scanning tools are discrete. They correspond to the 3D triangulated meshes known as the conventional representation of 3D surfaces. We will study here the extraction process of the novel representation on the meshes of the 3D faces.

4.1 3D mesh pre-processing

Many pre-processing steps are realized on the 3D faces meshes in order to ensure the best way to extract the 3PRL representation. The first step consists on removing non-connected parts of the 3D mesh. In fact, it is not possible to compute the geodesic distance between two points of the surface that belong to disconnected parts of a 3D mesh.

The second step consists on filling the holes on the surface. In order to have the same number of points for all faces and a finer resolution for both the level curves and the radial line ones, we make a remeshing procedure of the 3D meshes. A larger number of points is therefore obtained. All the pre-processing steps are computed by some functions provided with the VTK library (www.vtk.org).

4.2 Geodesic potential on 3D meshes

After the application of the pre-processing steps on the 3D meshes of faces, the geodesic potentials generated from the three reference points should be computed for the three-polar representation.

For a reference point, the corresponding geodesic potential consists on computing the geodesic distances between this point and each point of the 3D mesh. Several past methods have been proposed in the literature in order to compute distances on 3D meshes. We use in our work the Dijkstra algorithm [Dij59a] to compute the length of the geodesic path between two points of the surface.

4.3 Extraction of the level curves of the three polar representation

After computing the sum of the three geodesic potentials generated from the corresponding reference points, we should extract the level curves of the three-polar representation. In practice, the determination of a curve of level λ of the sum of the three geodesic potentials consists on the extraction of a trip rather than a curve. It corresponds to the set of surface points that have this sum values ($U_3 = U_{r_1} + U_{r_2} + U_{r_3}$) in the interval $[\lambda - \varepsilon, \lambda + \varepsilon]$. ε is a small positive real value.

4.4 Extraction of the radial line curves

The radial line curves of a surface S according to a starting point q consist on the intersection between the planes with the same angular separation and the surface. Since the 3D surface is composed by a 3D mesh, a plane does not slice the surface necessary on points (the plane could pass through the edges of triangles). Therefore, a radial curve R_q^α will be composed by a set of points of S that have Euclidean distances to the plane P_q^α less than a small positive real value ε_p.

$$R_q^\alpha = \{p \in S; de(p, P_q^\alpha) \leq \varepsilon_p\}. \quad (7)$$

with $de(p,q)$ is the Euclidean distance between two points p and q. P_q^α is the plane passing by the point q with an angular separation value equal to α according to the reference plane P_q^0. We note that the Euclidean distance between a point and a plane consists on the one between the point and its orthogonal projection on the same plane. The Fig. 1 illustrates the extraction procedure of the radial line curves on 3D meshes.

For the case of the 3D faces, the reference radial line curve corresponds to the vertical one when the face is moved to the upright position.

5 EFFECTIVENESS OF THE NOVEL REPRESENTATION FOR 3D FACES DESCRIPTION

We use in our experimentation the 3D faces data base BU-3DFE (Binghamton University - 3D Facial Expression) [Lij06]. This database contains a total of 100 subjects composed by 56 women and 44 men. For each subject, seven facial expressions are performed (neutral, disgust, happiness, angry, surprise, sadness and fear).

For the 3PRL representation, we used a total of 30 faces corresponding to six subjects of the database (3 men and 3 women). Five facial expressions are chosen for each subject including the neutral expression. We use for the construction of the three-polar representation the three reference points corresponding to the two outer corners of eyes and the nose tip [Jri14] (Fig. 2(a)). The level 0 of the sum of the three-polar representation can be composed by one or multiple points. We compute its centroid. This point will be denoted by C_e. All the level curves of the three-polar representation can be seen as curves around this point (Fig. 2(b)). C_e will be also the starting point of the radial line curves with the same angular separation (Fig. 2(c)). The Fig. 2(d) shows the invariant points of the novel 3PRL representation extracted from a 3D

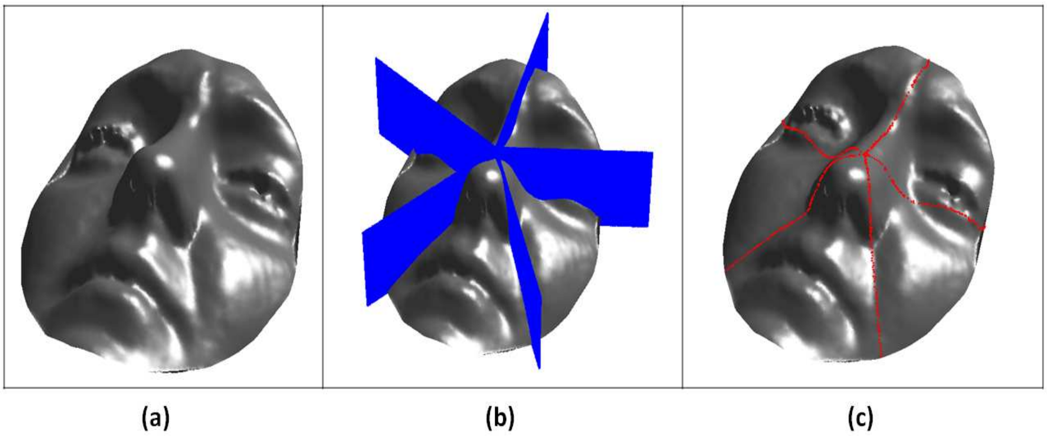


Figure 1: The extraction of radial line curves from a 3D face mesh: (a) a 3D face. (b) the planes with the same angular separation (blue color) and the face. (c) the radial line curves extracted from the surface.

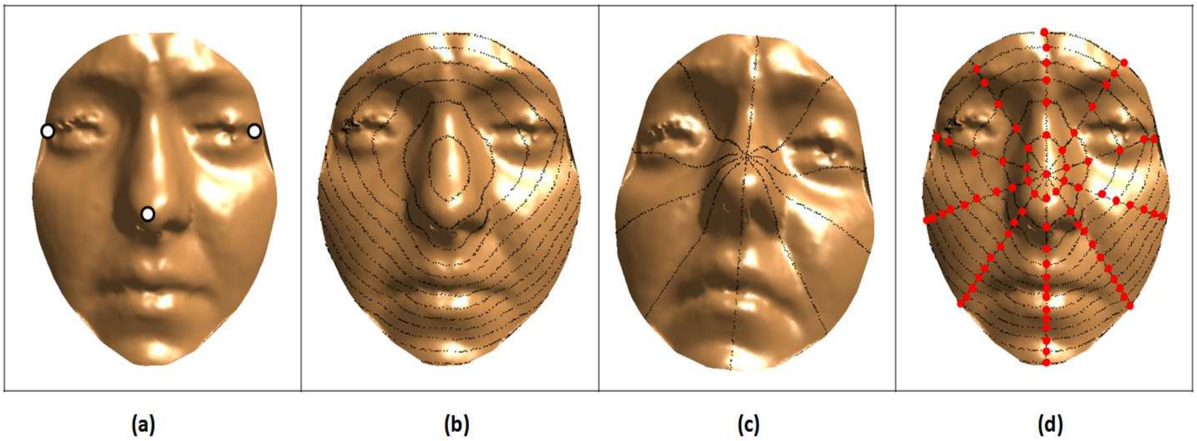


Figure 2: (a) The used three reference points. (b) the three-polar representation. (c) The radial line curves. (d) The 3PRL representation (invariant points with the red color).

face mesh.

In order to test the effectiveness of the novel representation to describe 3D faces, the Hausdorff shape distance is computed between each pair of invariant points cloud of faces representations. we obtain therefore a matrix of normalized Hausdorff shape distance. The Fig. 3 illustrates this matrix.

The table 1 summarizes the organization of the faces in this matrix.

From the observation of this matrix of distances, we can note that the novel invariant representation well describes the 3D faces. Indeed, the distances between the faces of the same person with different facial expressions are smaller than the other distances.

We make also an experimentation for the retrieval process. The Fig. 4 illustrates the obtained results.

The first column corresponds to the query subjects and the retrieved results are presented in the rest of this table. The query subjects are chosen to be the neutral faces of the six persons. From the explanation of the obtained results, we can note the effectiveness of such novel representation for the retrieval procedure. In fact only two errors exist in the retrieval process (faces with red color squares in the table.)

6 CONCLUSION

In this paper, we have introduced a novel 3D surfaces representations that we called 3PRL. It is obtained by combining two main representations: the three-polar representation and the one defined by the radial line curves from a starting point. The 3PRL representation consists on the set of invariant points corresponding to the intersection between the two last representations. The approximation of such representation on the 3D discrete meshes was studied. Its accuracy for 3D faces description and retrieval was evaluated.

The perspectives of such work are multiple. We, first,

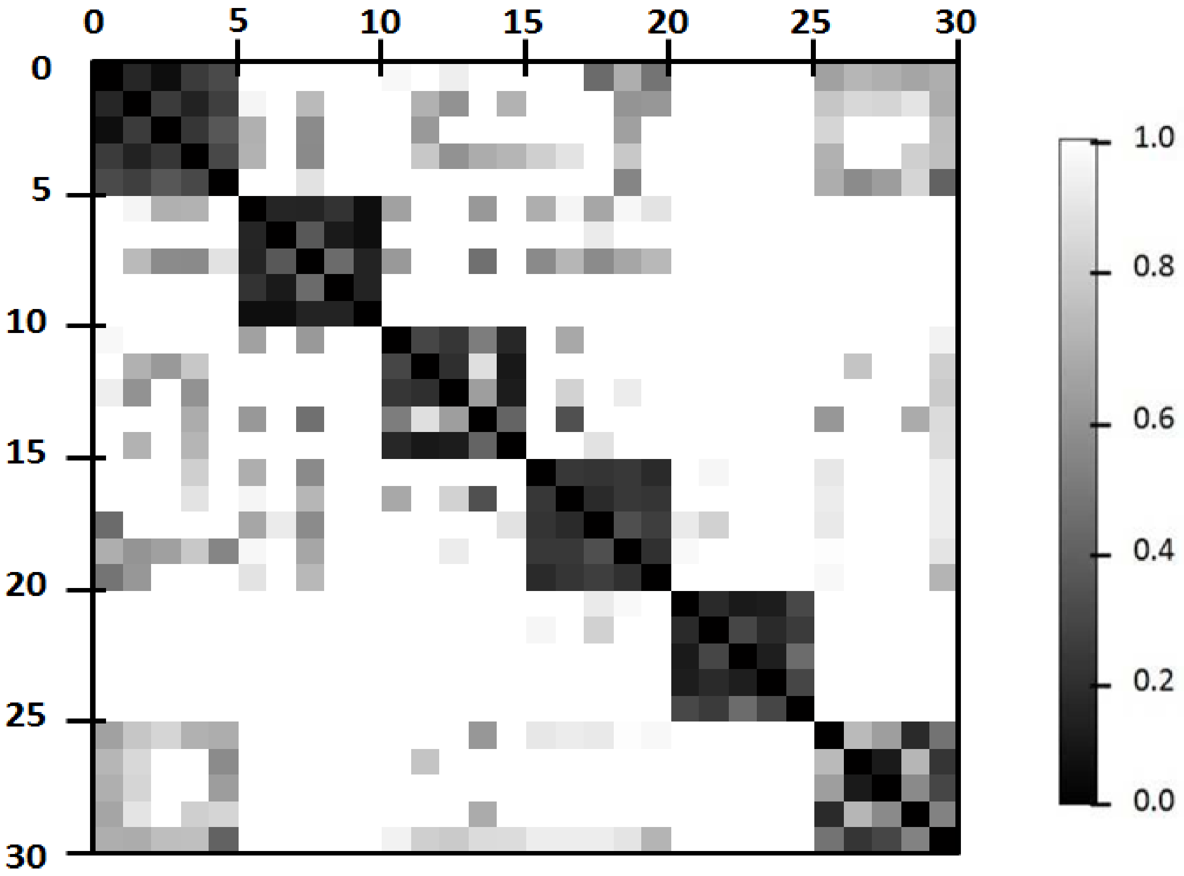


Figure 3: Matrix of pairwise normalized Hausdorff shape distance

Rows 1..5 Columns 1..5	Rows 6..10 Columns 6..10	Rows 11..15 Columns 11..15	Rows 16..20 Columns 16..20	Rows 21..25 Columns 21..25	Rows 26..30 Columns 26..30
Five faces of the subject 1	Five faces of the subject 2	Five faces of the subject 3	Five faces of the subject 4	Five faces of the subject 5	Five faces of the subject 6

Table 1: Organization of the data in the matrix

propose to experiment this novel representation on a larger number of faces. We intend also to make a study of the optimal numbers of levels of the three-polar representation and of the used radial line curves. It will be also interesting to define the optimal number of the reference points.

7 REFERENCES

[Ban00] Bannour, M.T., and Ghorbel, F. Isotropie de la représentation des surfaces; Application à la description et la visualisation d'objets 3D, in Conf.proc. RFIA 2000, pp. 275-282, 2000.

[Bes92] Besl,P.J., and Mckay, N.D. A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, No 2, pp. 239-256, 1992.

[Bur92] Burdin, V., Ghorbel, F., Tocnaye, J.D.B.D.L., and Roux, C. A three-dimensional primitive extraction of long bones obtained from bi-dimensional Fourier descriptors, Pattern Recognition Letters, vol. 13, No 3, pp. 213-217, 1992.

[Che03] Chen, D.Y., Tian, X.P., Shen, Y.T., and Ouhyoung, M. On Visual Similarity Based 3D Model Retrieval, Computer Graphics Forum, vol. 22, No 3, pp. 223-232, 2003.

[Dar04] Daras, P., Zarpalas, D., Tzovaras, D., and Strintzis, M.G. Shape Matching Using the 3D Radon Transform, in Conf.proc. Second International Symposium 3D Data Processing, Visualization, and Transmission, pp. 953-960, 2004.

[Dij59a] Dijkstra, E.W. A note on two problems in connection with graphs, Numerische Mathematik, 1959.

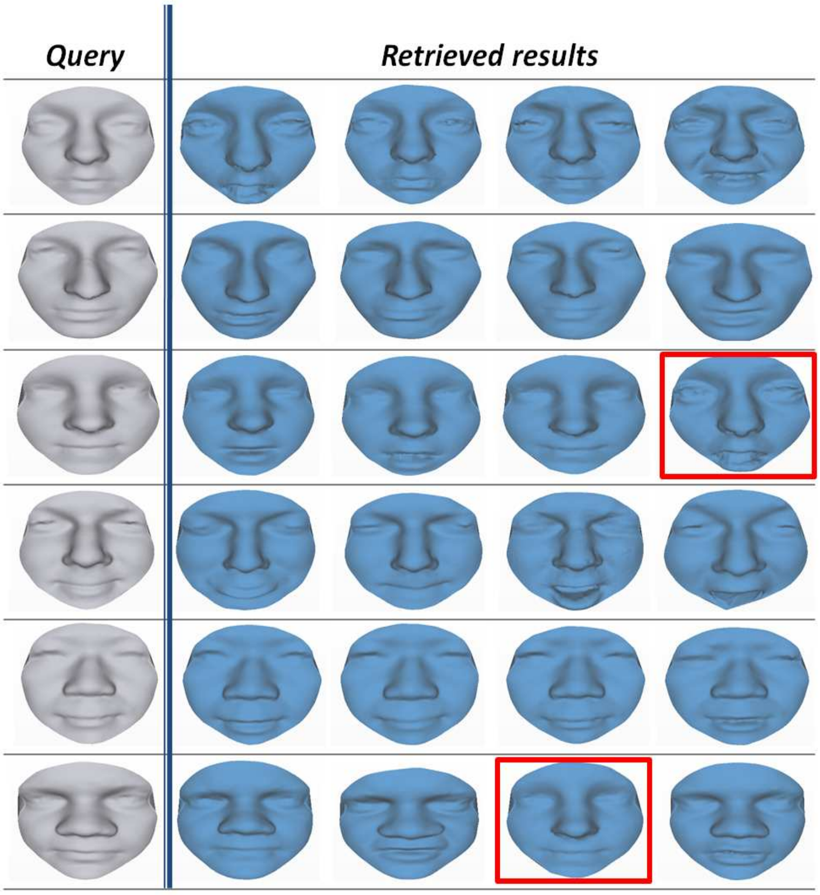


Figure 4: The retrieved results from the BU-3DFE database

[Fau86] Faugeras, O.D., and Hebert, M. The representation, recognition and positioning of 3D shapes from range data, techniques for 3D machine perception, Edition A, Rosenfield, Hollande, 1986.

[Gad12] Gadacha, W., and Ghorbel, F. A new 3D surface registration approach depending on a suited resolution: Application to 3D faces, in conf. proc. IEEE Mediterranean and Electrotechnical Conference (MELECON), Hammamet, Tunisia, 2012.

[Gho12] Ghorbel, F. Invariants for shapes and movement. Eleven cases from 1D to 4D and from euclidean to projectives (French version), Arts-pi Edition, Tunisia, 2012.

[Gho98] Ghorbel, F. A unitary formulation for invariant image description: application to image coding, special issue Annales des telecommunications, vol. 53, No 5-6, pp. 242-260, 1998.

[Jri13] Jribi, M., and Ghorbel, F. An Invariant Three-polar Representation for R^3 Surfaces: Robustness and Accuracy for 3D Faces Description, In Proc. the International Conference on Systems, Control, Signal Processing and Informatics. SCSI'13, 2013.

[Jri14] Jribi, M., and Ghorbel, F. A Stable and Invariant Three-polar Surface Representation: Application to 3D Face Description, In Proc. WSCG'14, the 22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Republic, 2014.

[Kaz03] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors, in Conf.proc. Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 156-164, 2003.

[Khe08] Bel Hadj Khelifa, W., Ben Abdallah, A. and Ghorbel, F. Three dimensional modeling of the left ventricle of the heart using spherical harmonic analysis, in Conf.proc. 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2008), Paris, France, 2008.

[Lag06] Laga, H., Takahashi, H., and Nakajima, M. Spherical Wavelet Descriptors for Content-Based 3D Model Retrieval, in Conf.proc. IEEE International Conference on Shape Modeling and Applications, pp. 15-25, 2006.

[Lij06] Lijun, Y., Xiaozhou, W., Yi, S., Jun, W., and Matthew, J., A 3D Facial Expression Database For Facial Behavior Research, in Conf.proc 7^{th} International Conference on Automatic Face and Gesture Recognition, pp. 211 - 216, 2006.

[Ric05] Ricard, J., Coeurjolly, D., and Baskurt, A. Generalizations of Angular Radial Transform for 2D and 3D Shape Retrieval, Pattern Recognition Letters, vol. 26, No 14, pp. 2174-2186, 2005.

[Sam06] Samir, C., Srivastava, A., and Daoudi, M. Three dimensional face recognition using shapes of facial curves , IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, No 11, pp. 1858-1863, 2006.

[Sri08] Srivastava, A., Samir, C., Joshi, S.H., and Daoudi, M. Elastic shape models for face anlysis using curvilinear coordinates, Journal of Mathematical Imaging and Vision, vol. 33, No 2, pp. 253-265, 2008.

[Sun03] Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton based Shape Matching and Retrieval, Shape Modeling International 2003, p. 130, 2003.

[Tun05] Tung, T., and Schmitt, F. The Augmented Multiresolution Reeb Graph Approach for Content-Based Retrieval of 3D Shapes, International Journal of Shape Modeling, vol. 11, No 1, pp. 91-120, 2005.

[Vra04] Vranic, D.V. 3D Model Retrieval.PhD dissertation, University Of Leipzig, 2004.

Analysis of 3D Mesh Correspondences Concerning Foldovers

Johannes Merz
TU Darmstadt
Darmstadt,
Germany
johannes.merz@
gris.tu-darmstadt.de

Roman Getto
TU Darmstadt
Darmstadt,
Germany
roman.getto@
gris.tu-darmstadt.de

Tatiana von Landesberger
TU Darmstadt
Darmstadt,
Germany
tatiana.von.landesberger@
gris.tu-darmstadt.de

Dieter W. Fellner
TU Darmstadt &
Fraunhofer IGD,
Darmstadt,
Germany
dieter.fellner@
gris.tu-darmstadt.de

ABSTRACT

Foldovers (i.e., folding of triangles in a 3D mesh) are artifacts that cause problems for morphing. Mesh morphing uses vertex correspondences among the source and the target mesh to define the morphing path. Although there exist techniques for making a foldover-free mesh morphing, identification and correction of foldovers in existing correspondences is still an unsolved issue.
This paper proposes a new technique for the identification and resolution of foldovers for mesh morphing using predefined 3D mesh correspondences. The technique is evaluated on several different meshes with given correspondences. The mesh examples comprise both real medical data and synthetically deformed meshes. We also present various possible usage scenarios of the new algorithm, showing its benefit for the analysis and comparison of mesh correspondences with respect to foldover problems.

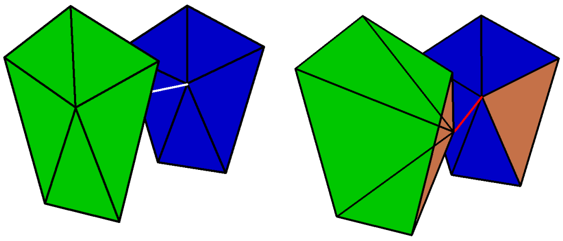
Keywords

Foldover, Correspondence, Morphing, Mesh Comparison

1 INTRODUCTION

Mesh morphing is commonly used in various computer graphic applications for interactively showing animated correspondence between two meshes – the source mesh and the target mesh. Mesh morphing can be described as the continuous deformation from a graphical object to another one [Gom99]. During the deformation the points from the first mesh move to their corresponding points on the second mesh along the correspondence path.

Morphing methods using linear paths between mesh correspondences can show problems if the source triangle folds over during the morphing process. It means that a triangle t with a positively-oriented area is mapped to a triangle $f(t)$ with a negatively-oriented area [Ebk+13] (see Figure 1). Foldovers often occur, for example, when morphing between complex meshes like an automatically segmented and expert-segmented liver from 3D medical images. Foldovers can produce unnaturally looking morphing sequences. Moreover, the foldover problem can lead to undesirable blending of neighboring textures when using texture mapping algorithms. Therefore, we need to identify and to resolve such foldovers.



(a) No foldover. (b) Foldover.

Figure 1: **(a):** no foldover. During the morphing from the source mesh (blue) to the target mesh (green) the center point does not leave the neighboring region. However in **(b)** this is the case, causing the positive area of the brown triangle to become negative: a foldover.

Mesh morphing may rely on point-to-point mesh correspondences, which are used for defining morphing paths. There are various methods for defining mesh correspondences [Tam+13; Tam+14; van+11]. One possibility is to use the correspondences determined by local mesh distance measures such as the surface distance or the extended surface distance [CRS98; GJC01; GKL15] . The (Extended) Surface Distance relates one point of the first mesh to one point of the second mesh and thus creates correspondences. However, not all distance measures can be used for determining correspondences. For example, the Fréchet Distance [VH01] is based on a parameterization and thus requires preknown correspondences.

Depending on the method of determining the correspondences, the points on the meshes that are associated with each other may differ and often the quality is application-dependent. The differences between sets of correspondences can lead to qualitatively different morphing results. Thus, we also need to analyze and to compare correspondences with regard to foldover problems.

We would like to emphasize, that we focus on identification and solution of foldovers for predefined correspondences. Although there are techniques that can produce foldover-free mesh morphing [MZX14] they cannot use the existing correspondences. The usage of correspondences, however, is required in many cases such as the assessment of segmentation quality or mesh registration quality in 3D medical image segementation applications. As current foldover identification methods are not suitable for such 3D mesh correspondence-based problems (see Section 2), the identification and correction of foldovers for a given set of correspondences remains an open research question.

Contribution & Application Benefit

We propose a new technique for identifying foldovers given a mesh and correspondences to another mesh (see Section 3). This enables the evaluation of correspondences concerning foldovers as well as the comparison of different sets of correspondences for a given mesh. Moreover, we automatically determine corrections to the foldover errors. The hereby obtained correction can be used to draw inferences about the used correspondence estimation method.

Our method can be applied to the analysis of 3D mesh correspondences in the evaluation of 3D medical image segmentation results, to the evaluation of surface distances creating correspondences, to the assessment of mesh deformations or to enable texture-enabled morphing using predefined correspondences. In the evaluation section, we show the benefit of our method for the first two usage scenarios (see Section 4).

2 RELATED WORK

This section discusses related work in the area of foldover handling. Foldover handling is most present in areas such as mesh morphing, texture mapping or mesh merging.

Regarding mesh morphing, Fujimura and Makarov [FM98] developed a technique for foldover-free image warping. They propose an approach, that uses a time-dependent triangulation of the image. The triangulation of the image is changed, when the foldover occurs. This happens, when a point leaves its surrounding polygon during warping. At this point in time the structure of the triangulation changes by removing unneeded edges and inserting new ones. Figure 2 shows an example of their approach. The moments of the foldover events can be precalculated by computing the intersection of the correspondence path. Although it produces foldover-free warping despite of erroneous correspondences, it does not represent a method to detect and correct foldovers in existing correspondences. This merely avoids foldovers. Furthermore it is developed for 2D images, not 3D meshes.

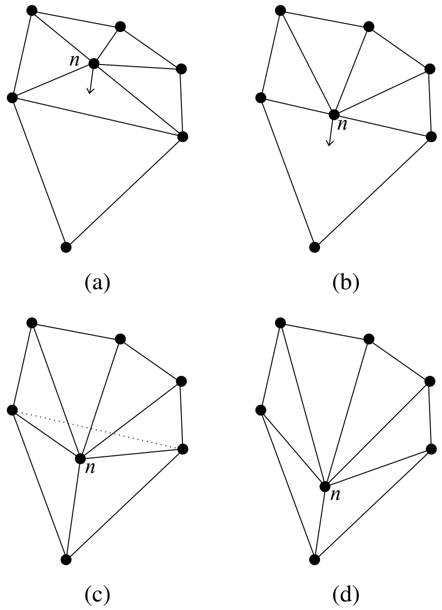


Figure 2: During the image warping the point n moves along its correspondence path and leaves its surrounding polygon. A foldover occurs.

Lee et al. [LYY08] used a similar approach for addressing the foldover problem in texture mapping. To resolve foldovers they also use edge swaps at precalculatable moments.

Also working on foldover-free image warping, Ma et al. [MZX14] presented a technique, that maintains the triangulation of the source mesh. Instead of precalculating the foldover events along the correspondence path, their technique uses the help of radial basis functions (RBF). The first main step is to prepare the trajectories (correspondence paths) $\mathbf{C}_i$ for the warping. The $\mathbf{C}_i$ are piecewise linear polylines that are constructed iteratively out of the nodes $C_i^j, j = 0, 1, \ldots, m$, where C_i^0 is the source point and C_i^m is the corresponding point: $\{C_i^0\} \to \{C_i^1\} \to \cdots \to \{C_i^m\}$. Moreover they may not intersect. Following, the interpolation is executed over the span $\{C_i^j\} \to \{C_i^{j+1}\}$ with a stepsize

$\delta^{j,l}$. To ensure local bijectivity, the stepsize has to be a compromise of the RBF stepsize and the maximum stepsize until a foldover occurs. Other than in [FM98] this method does not adapt the triangulation to avoid foldovers, but the correspondence paths. Thus it also avoids foldovers during the warping process and is no method to analyze the data and perform corrections. Furthermore it cannot be precalculated.

Mocanu et al. [MTT13] developed a technique for mesh deformations that also makes use of radial basis functions. They observed, that an increased number of intermediate steps reduces the amount of foldovers. Similar to the previously mentioned techniques it aims at avoiding foldover rather than correcting it as well.

Looking into mesh merging, Alexa addressed the foldover problem [Ale00; Ale02]. He proposed an algorithm to merge genus 0 (without holes) meshes. While merging meshes, it is possible to introduce foldovers in the resulting mesh. The following function is used for the mapping:

$$f(x) = \begin{cases} \mathbf{x} + c(d - \|\mathbf{x}-\mathbf{v}\|)(\mathbf{w}-\mathbf{v}) & \|\mathbf{x}-\mathbf{v}\| < d \\ \mathbf{x} & \|\mathbf{x}-\mathbf{v}\| \geq d \end{cases} .$$

The function describes a point $\mathbf{x}$ moving from $\mathbf{v}$ to $\mathbf{w}$. If the mapping results in foldovers, the factor $c = 0.5$ can be adapted to modify the influence of both meshes on the result. Altering d changes the influence radius, resulting in smaller steps. Although the proposed algorithm works with 3D meshes, it again aims at avoiding foldovers during the merging process.

The discussed work proposes multiple ways of handling the foldover problem, but none focuses on identifying existing foldovers and suggesting a corrected correspondence or works on 3D meshes.

3 APPROACH

3.1 Overview

The main idea of our approach is to extend the algorithm of [FM98] for detecting error points (i.e. foldovers) in three-dimensional space. We extend the algorithm of Fujimura as it is a one-step technique. However, its extension to 3D is not trivial. We need to solve several problems, such as possible rotations, skew lines or non-planarity of quads.

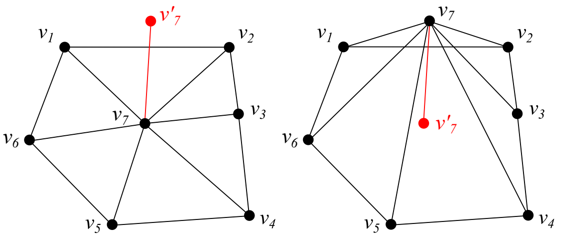
The core of our approach is the detection of error points leading to foldovers and their correction. This core approach is described in Section 3.2. As some error points detected by this algorithms can be false positives (e.g., due to rotations or non-planarity), we need to avoid them. For this purpose, we present several algorithms detecting these problems, thus extending our core approach (see Section 3.3). This also includes an extended error correction using weighting of errors.

3.2 Core Approach

In this section, we describe our core algorithms for detecting and correcting foldover error points.

3.2.1 *Detection of Error Points*

Fujimura and Makarov described the detection of foldover events in two-dimensional meshes in [FM98]. They state that the problem appears when a point leaves the neighboring polygon while being moved along its correspondence path (see Figure 3).



(a) Source mesh. (b) Target mesh.

Figure 3: We show a two-dimensional polygon with a translation of the point v_7. From **(a)** to **(b)** v_7 moves along its correspondence path outside of the polygon. Hence, the correspondence intersects the edge $v_1 - v_2$ of the polygon.

We use the same concept in 3D. For a foldover to be detected, the correspondence path no longer needs to intersect an edge of the points neighboring region. The correspondence path $cp(t)$ rather has to intersect one of the sides S of the area between the source and the target area (see Figure 4). In this case, the area of the affected triangle becomes negative:

$$\exists s \in S : (cp(t) - \vec{v_s}) \cdot \vec{n_s} = 0 \quad , t \in [0,1].$$

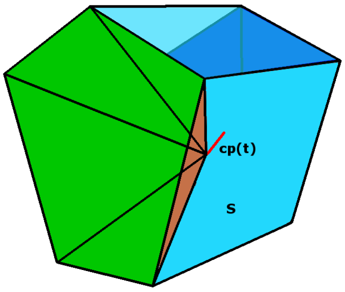
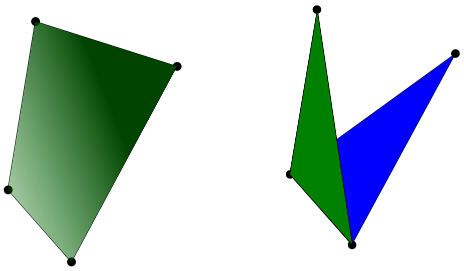


Figure 4: In the case of a foldover, the correspondence path $cp(t)$ intersects one of the sides S (light blue) of the area between the source mesh (blue) and the target mesh (green).

Each side is put up by two source points and two target points. A surface that contains four points in three-dimensional space does not need to be planar, which makes it hard to calculate intersections. Therefore, the sides are approximated by two planes, each using the

```
for all $v \in V$ do
    $foldover \leftarrow false$
    for all $s \in S$ do
        if $cp(t)$ intersects $s_1$ or $s_2$ then
            $foldover \leftarrow true$
    if $foldover$ then
        $errors.append(v)$
```

Algorithm 1: Detect errors



(a) Non-planar quad.

(b) Approximation using two planes.

Figure 5: A quad that is build around four points does not need to be planar **(a)**. However with the help of two planes that are made out of three points each, this side can be approximated **(b)**.

two points from the source mesh $\mathcal{M}$ and one of the points from the target mesh $\widetilde{\mathcal{M}}$ (see Fig. 5):

$$s_1 : \vec{x} = (v_2 - v_1) + t\tilde{v}_1$$
$$s_2 : \vec{x} = (v_2 - v_1) + t\tilde{v}_2.$$

The core error detection algorithm is in Algorithm 1. The prerequisite for this technique is that the source mesh is 2-manifold. Otherwise the correspondence path may not intersect with any side even though a foldover exists. Furthermore the source mesh has to be foldover free, which should be ensured by a suitable parameterization. Holes in the mesh on the other hand do not interfere with the functionality, as the technique only depends on the direct neighbors of the current point and also works if the point is on the border of the mesh.

3.2.2 Correction of Error Points

With the detection Algorithm 1, it is possible to find vertices with correspondences that lead to foldovers. These problems are corrected by moving the target point to a valid area, so that the correspondence path does not leave the three-dimensional volume that is spanned by the neighboring region of the source and the target point and the area of the triangles that became negative turn positive again.

To satisfy this condition, various possible new positions can be used. Our algorithm moves the target point to the centroid c of its neighboring region with the points $[v_0, \dots, v_n]$:

$$c = \frac{\sum_{i=0}^{n} v_i}{n}$$

Using the centroid as the corrected point is not only easy to compute, it has the advantage that it has the optimal distance to the neighboring points. Thus the probability of a neighboring error affecting the current point is minimal. Algorithm 2 illustrates the sequences of steps for the correction.

```
for all $v \in errors$ do
    Find neighboring region $r$ on the target mesh $\widetilde{\mathcal{M}}$
    $correctedPoint \leftarrow r.ComputeCentroid()$
    $\tilde{v} \leftarrow correctedPoint$
```

Algorithm 2: Correct errors

3.3 Extension: Avoiding False Positives

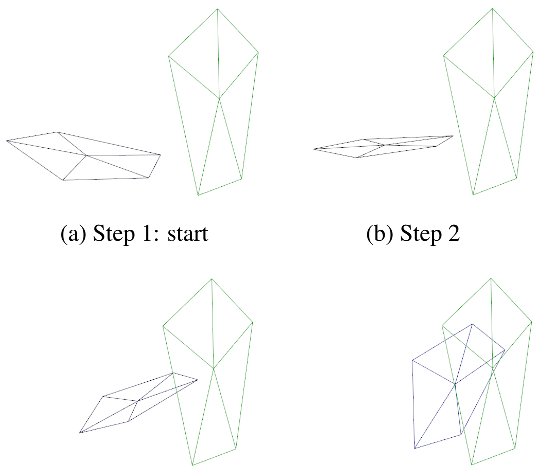
We extend our core error detection algorithm, as it may lead to false positives. For example rotation, mesh overlap or degenerated triangles can result in false positive error detections. In the following, these false positive errors are systematically filtered from the detected errors, leaving only relevant errors. Finally, the filtered errors are corrected using an extended correction algorithm, which orders the error correction according to the errors' severity.

3.3.1 Detecting Rotations

When comparing differently shaped meshes, it can happen that the corresponding regions on the source and the target mesh are rotated to one another by 90° or more. While morphing, the source region approaches the target region as shown in Figure 6.

Theoretically, the conditions for the detection of a foldover in rotated regions are the same as in the standard scenario described above. Nevertheless, the detection algorithm may mark a correct point as an error, because an intersection with one of the sides is identified. Figure 7a shows a scenario, where the detection algorithm incorrectly finds an error in a correct region, because the correspondence path crosses a side plane. In contrast, Figure 7c is an example for a correct error identification, despite of rotation. Hence both results are possible. To filter the false positive results, we first rotate the triangles before running the detection algorithm. Figures 7b and 7d show that the false positively detected error is resolved, whereas the correctly detected error remains. We perform the following three steps for this purpose:

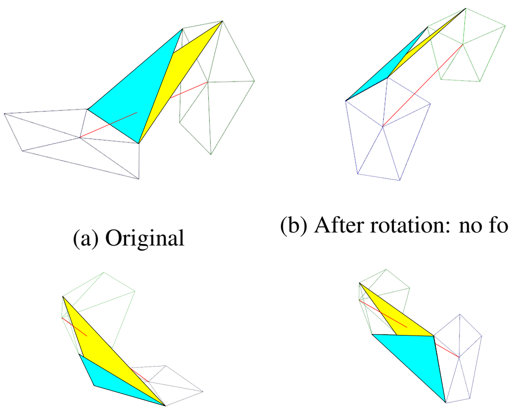
1. Approximate the surface normals of the source and target normals:
To determine the rotation matrix we need to calculate



(a) Step 1: start (b) Step 2

(c) Step 3 (d) Step 4: finish

Figure 6: The example shows how a region (blue) can rotate from the source mesh to a target mesh. In such a case a foldover does not have to occur.



(a) Original (b) After rotation: no foldover

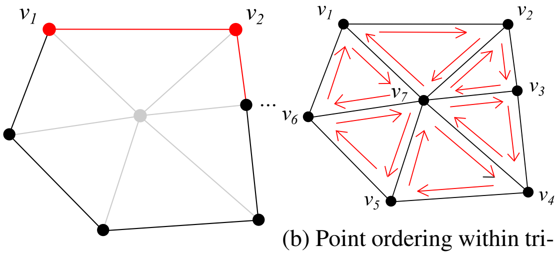
(c) Original (d) After rotation: foldover

Figure 7: Errors with and without rotation before the morphing. **(a)** and **(b)** represent a false positive error, whereas **(c)** and **(d)** show a real error.

the normals of both regions. With their help the rotation between two vectors can be computed. As the region that surrounds the current point is not a two-dimensional polygon, the surface consists of several triangles with different rotations in three-dimensional space. To determine one surface normal for the region, the surface normals of all triangles that are contained in the region are calculated and averaged. In our case, the surface normals were weighted by the area of the triangles. To ensure a correct addition of the single surface normals, they need to have the same orientation. This can be reached by ordering the points of the region in a constant manner. For this, we first discard all edges that contain the current point and then choose an arbitrary point. From this point, we iterate along the remaining edges to the next point and gain a sorted list of points (see Figure 8a). With the help of this list, we can now ensure a consistent orientation of all triangles. Each triangle starts at the center point and uses the previous and the next point in the sorted list (see Figure 8b). The surface normal N can now be calculated as follows:

$$\forall\triangle_i(v_1,v_2,v_3): N_i = (v_2-v_1)\times(v_3-v_1)$$
$$\Rightarrow N_{Region} = \frac{\sum_{i=0}^{n} N_i}{|\sum_{i=0}^{n} N_i|}.$$

2. Calculate the rotation matrix between normals:



(a) Overall point ordering (b) Point ordering within triangles.

Figure 8: To ensure that all surface normals have the same orientation, the points are ordered along the edges of the region **(a)**. Figure **(b)** shows the order of the points per triangle.

Now that we are able to calculate the surface normals of the source and the target region, it is possible to calculate the rotation matrix for a transformation from the source to the target normal. In three-dimensional space the rotation between two vectors can be described by three rotation angles α, β, γ. A way to retrieve them is to use quaternions, which are generalized complex numbers:

$$q = s + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = (\mathbf{v}, s).$$

Using unit quaternions $|\hat{\mathbf{q}}| = (\sin(\theta)r,\ \cos(\theta))$ it is possible to calculate the rotation matrix as:

$$\mathrm{R}(\theta, r) = \hat{\mathbf{q}}\hat{\mathbf{p}}\hat{\mathbf{q}}^{-1},$$

where $\hat{\mathbf{p}}$ is the unit quaternion of the point p that is to be rotated by θ around the axis r. According to Tomas et al. [AMHH08] this can be reduced to

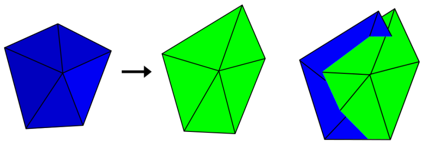
$$R(n_1,n_2) = \begin{pmatrix} e+hv_x^2 & hv_xv_y - v_z & hv_xv_z + v_y & 0 \\ hv_xv_y + v_z & e+hv_y^2 & hv_yv_z - v_x & 0 \\ hv_xv_z - v_y & hv_yv_z + v_x & e+hv_z^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where $v = n_1 \times n_2$, $e = n_1 \cdot n_2$ and $h = \frac{1}{1+e}$.

3. Execute detection Algorithm with transformed source region:
We now perform the detection algorithm 1 on the transformed region.

3.3.2 Detection of Overlaps, Sideward Movements and Degenerated Triangles

Similar to the problems with rotated regions, mesh overlap or sideward movement as seen in Figure 9 can lead to false diagnosis. Because of the side movement the angles of the side planes to the mesh regions are extremely sharp. Moreover the mesh overlap results in correspondence paths, that point in different directions. We use a similar technique as with the rotations. During the handling of the rotation we already calculated the normal vectors. Thus it is possible to translate all points of the source region in the direction of the target normal. The result is that the movement is not sidewards anymore, which prevents overlap during the movement.



(a) Before (b) After

Figure 9: These images illustrate a scenario, where a source region (blue) moves sidewards towards the target region (green). As a result of the three-dimensional structure of the meshes, they can overlap.

When creating mesh-correspondences not every source point is associated with a distinct target point. To the contrary, correspondences can be surjective. Such a many-to-one correspondence can create degenerated triangles if two points have the same correspondence. Furthermore a triangle can degenerate, when one point moves onto the egde between the other two points. Our algorithm tests, whether a triangle degenerates and discards an error that resulted from these conditions. This is done by calculating the distance between the correspondence points and between the points and the edges, respectively.

3.3.3 Correction using Error Weighting

Up to now, only isolated errors of one point were considered. The detection algorithm uses the neighboring points to span the side planes. If one of these neighboring points is itself faulty, the side planes lead to miscalculations of the detection algorithm. Thus a correctly detected error can imply false positive detections at the neighboring points as Figure 10 shows.
In order to distinguish the error categories, the influence of a detected error on the involved points is calculated. We define the influence of an error as the proportion the error has on the Euclidean distance $d(x,y)$ between the original correspondence point and the corrected correspondence point. This is possible, because a corrected correspondence point can even be calculated for false positive errors. Regarding Figure 10 error **C** has direct influence on the Euclidean distance between the original correspondence point and the corrected correspondence point, because it is the actual error. Both **A** and **B** are neighboring errors that result from error **C**. However they do not have a direct influence on the error, because the neighboring points that span the region around the error only have an influence of $\frac{1}{N}$, where N is the amount of neighboring points. Hence, it is possible to order the detected errors by their influence, also called error weight. Table 1 shows weights for Fig. 10.

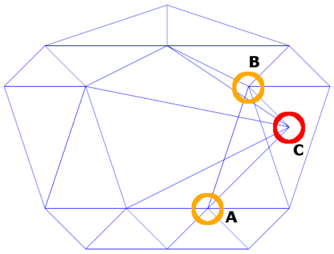


Figure 10: On the depicted target mesh, the point that results in a foldover is marked red. Due to this error correct neighboring points are mistakenly identified as errors. They are marked orange.

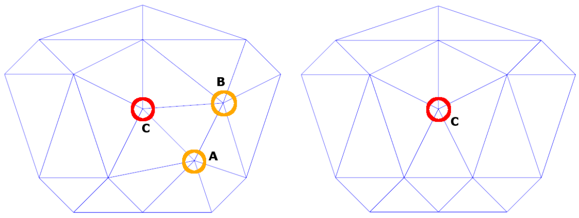
```
$errors.sort(DESCENDING)$
for all $v \in errors$ do
    if v.isStillError() then
        Find neighboring region $R$ on the target mesh $\widetilde{\mathcal{M}}$
        $correctPoint \leftarrow R.ComputeCentroid()$
        $\tilde{v} \leftarrow correctPoint$
    else
        $errors.remove(v)$
```

Algorithm 3: Correct errors in the order of their severity (i.e. error weight).

Error	Error Weight
C	22.50
B	2.01
A	1.25

Table 1: Errors from Figure 10 sorted in descending order by error weight. The actual error **C** is at the top of the list, before the resulting false positive errors.

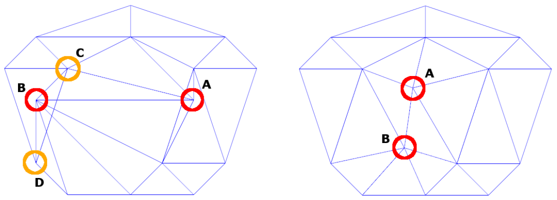
Then the correction algorithm corrects all errors. It starts with the first error in the list. The possibility for it to be before all its resulting false positive errors is very high. Hence, repeating the test for errors should yield no more false positive results. The correction algorithm can thus be extended to retest each error prior to the correction as described in Algorithm 3. The effect of the error ordering by error severity (error weights) can be seen in Figure 11.



(a) Without weighting. (b) With weighting.

Figure 11: Figure **(a)** shows the result of the correction without error weighting. All errors are corrected, regardless of whether they were actual errors or false positives. With weight-based ordering, **(b)** only the correspondence point that actually leads to a foldover is corrected.

The previous example showed false positive errors that result from neighboring real errors. However, the correction using error weighting also works with neighboring real errors. Figure 12 shows the correction of neighboring real errors. The resulting false positive errors are hereby filtered by the error weighting.



(a) Before the correction. (b) Weighted correction.

Figure 12: In **(a)** two neighboring real errors are marked red. The resulting false positive errors are marked yellow. **(b)** shows the example after the correction with error weighting. Only the real errors were corrected.

4 EVALUATION

With the help of several example meshes, we evaluate the analysis of mesh correspondences with regard to foldovers. We also discuss the advantages and demonstrate possible applications of the analysis as well as the limitations.

We present foldover analysis of two sets of data. First, correspondences of automatic and expert 3D medical image segmentations, and, second, correspondences between original and deformed 3D meshes. Note that we used simple not foldover-free correspondence estimation methods for the evaluation of our approach.

The first case analyzes foldovers in the correspondences resulting from the comparison of two meshes: an automatic 3D medical image segmentation and a segmentation performed by an expert. As medical image segmentation experts are interested in detecting segmentation errors (i.e., large distances between the automatic and expert segmentations), we use local distances for defining mesh correspondences. In particular, we employ and compare two distance measures: a commonly used Surface Distance [GJC01] and its recent improvement – the Extended Surface Distance [GKL15]. We analyzed the correspondences of segmentations of six livers and one carotid artery. The data stems from real images.

Second use case analyzes foldovers in the correspondences between original and deformed banchmark datasets: The Stanford Bunny (http://graphics.stanford.edu/data/3Dscanrep/), the Frog and the Buste (Both http://www.aimatshape.net/).

Table 2 shows the amount of vertices and triangles of the data sets.

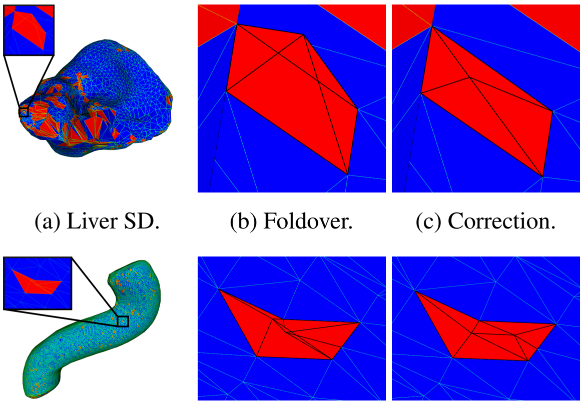
Example	# Vertices	# Triangles
Liver 1	2562	5120
Liver 2	6002	12000
Liver 3	3996	8000
Liver 4	4000	8000
Liver 5	4002	8000
Liver 6	3996	8000
Carotid artery	30674	61344
Bunny	2533	5062
Frog	5002	10000
Buste	5002	10000

Table 2: The table shows the amount of vertices and triangles of the organ segmentation examples.

The presented foldover analysis and correction technique aims at detecting existing correspondences that result in foldovers and correcting them by moving the correspondence to a foldover free position.

To evaluate the technique, we first look at the identified foldover errors in the data visualizations. Figures 13a - 13c show the foldover result of a liver mesh comparison using SD. An isolated error is highlighted by marking the neighboring region red. Figure 13b clearly shows that, prior to correction, the correspondence point lies outside of the triangulation of the neighboring points, thus results in a foldover. After the correction, the erroneous correspondence point is moved inside the valid area (see Figure 13c).
Analogously, Figures 13d - 13f display the same scenario on the artery mesh using SD.

For deeper evaluation, Figure 14 displays two cases with large areas, where many errors exist. Both examples show the area before and after the correction is executed.
Figure 15 shows a comparison of a morphing sequence with foldover correction to one without foldover correction using ESD. In the uncorrected morphing it is visi-



(a) Liver SD. (b) Foldover. (c) Correction.

(d) Artery SD. (e) Foldover. (f) Correction.

Figure 13: After the analysis of the SD of a liver mesh **(a)** and an artery mesh **(d)**, the errors are marked red. Pictures **(b)** and **(e)** show obvious foldovers. In **c)** and **(f)** they are corrected.

ble, that during the sequence foldovers occur. The corrected morphing sequence stays foldover-free, without harming the global structure of the correspondences.

Alongside the sole correction of foldover problems, the analysis was also developed to be able to assess different sets of correspondences concerning their degree of foldover or characteristic foldover regions and compare them with others. In the following example (see Figure 16) a mesh of an automatic liver segmentation is shown. Marked with yellow is the region with the most detail of the mesh. It represents the porta hepatis (entry of artery and vein) of the liver. It is noticeable that the analysis detects several errors forming a ring around the area. This indicates, that the correspondences are prone to foldovers in high detail areas.

By now, we have used the foldover analysis to evaluate a given correspondence. Now, we look at different sets of correspondences and compare them with the help of the foldover analysis. Table 3 shows the results of the foldover comparison of SD and ESD on five different liver segmentation data. We report the amount of errors detected as well as their percentage in all triangles. This evaluation provides us with the insight, that the ESD produces more foldover problems than the SD. However the result is only valid for liver meshes, as different correspondences perform differently on varying mesh structures.

5 LIMITATIONS

Our approach corrects the found errors by moving the faulty correspondence points and thus resolving the foldover. This results in a slight change of the position of the corresponding point in the mesh. Because of the fact that the target mesh is constructed out of points on the second mesh of the comparison with the triangulation of the first mesh, the corrected correspondences do not necessarily lie on the second mesh. However, the correction is designed to draw inferences about the correspondence estimation method. It could thus be used as an indicator for the method. Moreover, the correspondence estimation can easily be repeated after the correction, to iteratively find foldover-free correspondences that completely lie on the second mesh.

Furthermore it must be noted that the correction algorithm only corrects foldover errors. If the underlying correspondences are not expressing a meaningful relation between the two meshes, the algorithm cannot improve that, but merely make it foldover-free.

The algorithm was also not designed to detect foldovers that were introduced through self collision of the mesh. Even if some parts of the mesh touch during morphing this is not a foldover, as the triangulation does not define the overlapping points as neighboring.

Due to inaccuracies in some parts of the algorithm a small amount of false positive errors can still remain. For example the approximation of the surface normal can be inaccurate, if one triangle of the region is very different from the rest. Additionally the computation of the intersections and the test for degenerated triangles works with an ε-precision. Depending on the chosen ε the amount of remaining false positive errors can be adjusted. To reduce this amount the algorithm can be repeated by reanalyzing the corrected correspondences. Table 4 illustrates this issue.

	Surface Distance		**Extended S. Distance**	
Liver	# Errors	%	# Errors	%
1	60	2,342%	122	4,762%
2	162	2,7%	187	3,12%
3	122	3,053%	118	2,953%
4	145	3,363%	134	3,35%
5	122	3,048%	194	4,85%
ø		**2,9012%**		**3,807%**

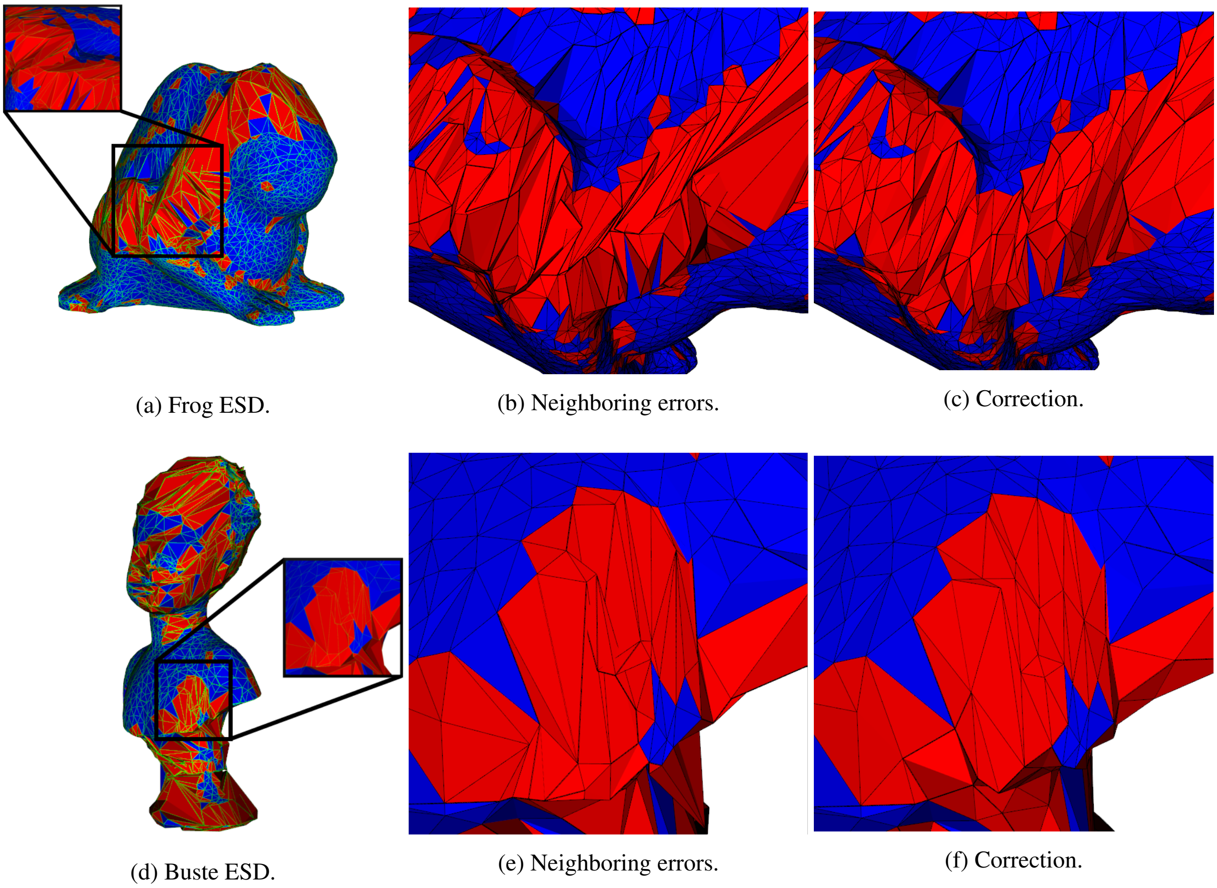
Table 3: Based on the five examples, the probability of a foldover in the category of liver segmentations of SD compared to ESD is evaluated.

	# Errors	
ε	$\mathcal{Z}$	$\mathcal{Z}_{\mathcal{K}}$
10^{-2}	27	21
10^{-3}	59	53
10^{-4}	62	59

Table 4: Depending on the chosen ε-precision the amount of leftover false positive errors changes. By repeating the algorithm it can be minimized. $\mathcal{Z}$ is the result of the first and $\mathcal{Z}_{\mathcal{K}}$ the output of the second correction.

6 CONCLUSION & FUTURE WORK

We proposed a new method to identify foldovers in 3D mesh correspondences. The method also offers a cor-



(a) Frog ESD.

(b) Neighboring errors.

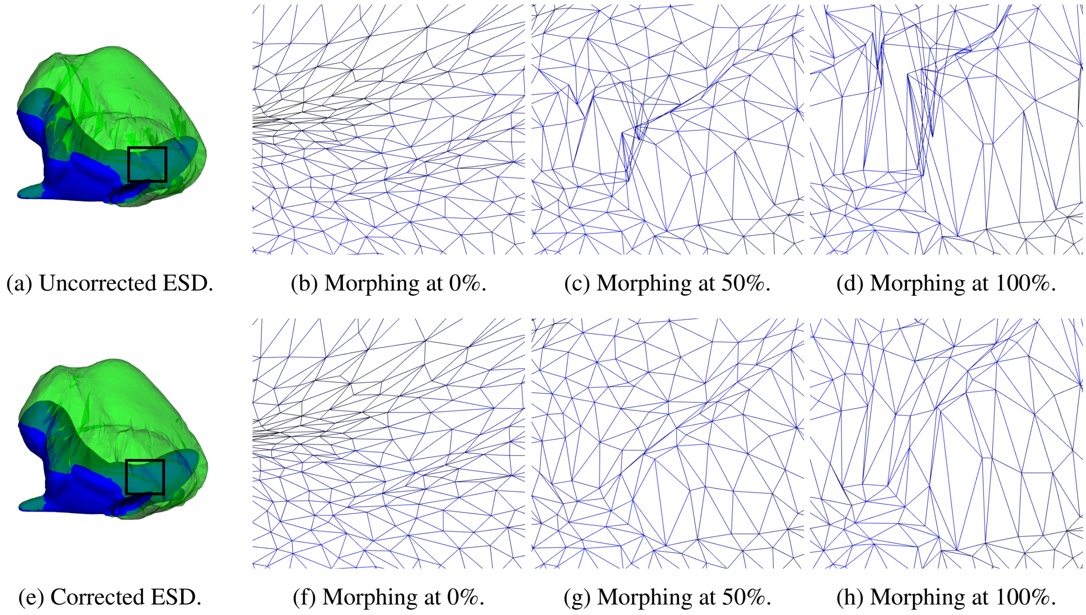
(c) Correction.

(d) Buste ESD.

(e) Neighboring errors.

(f) Correction.

Figure 14: ESD of the mesh of the frog **(a)** and the buste **(d)** are analyzed and corrected.



(a) Uncorrected ESD.

(b) Morphing at 0%.

(c) Morphing at 50%.

(d) Morphing at 100%.

(e) Corrected ESD.

(f) Morphing at 0%.

(g) Morphing at 50%.

(h) Morphing at 100%.

Figure 15: This figure compares a morphing sequence without foldover correction (**(a)** - **(d)**) to one with foldover correction (**(e)** - **(h)**) using ESD. The global structure of the correspondences is not harmed by the correction. For better visibility only the morphed mesh is shown in the steps, leaving out the source and target mesh.

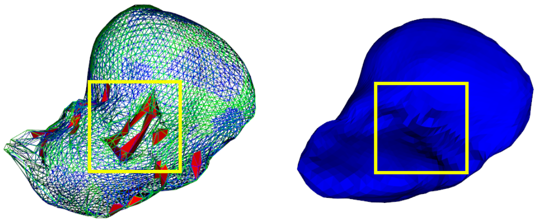


Figure 16: Around the area with the most detail of the mesh, an increased number of foldovers occur.

rection of the correspondences. It extends the work of Fujimura and Makarov [FM98] to 3D meshes and avoids false positive error detections.

The proposed algorithm is able to detect single and multiple foldover errors. Furthermore the correction provides a meaningful suggestion for the correspondences, provided the correspondences are of sufficient quality. With the obtained information it is possible to evaluate the given correspondences and compare them to others. The detection of false positive errors could be further improved with more reliable surface normals, as the surface normal calculation is not stable to outliers.

In the future, we will include the analysis into the development of a new distance measure, to ensure foldover-free correspondences during their generation. Moreover the amount of approximation regarding the side planes and the rotation will be reduced.

7 ACKNOWLEDGMENTS

The authors would like to thank Meike Becker, Matthias Kirschner and Georgios Sakas at TU Darmstadt as well as Stefan Wesarg at Fraunhofer IGD for providing us with the 3D medical image segmentation data and helping with the evaluation. The work was supported by DFG within a SPP 1335 project.

8 REFERENCES

[Ale00] Marc Alexa. "Merging polyhedral shapes with scattered features". English. In: *The Visual Computer* 16.1 (2000), pp. 26–37.

[Ale02] Marc Alexa. "Recent advances in mesh morphing". In: *Computer Graphics Forum*. Vol. 21. 2. Wiley Online Library. 2002, pp. 173–198.

[AMHH08] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-Time Rendering, Third Edition*. Taylor & Francis, 2008. ISBN: 9781439865293.

[CRS98] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. "Metro: Measuring error on simplified surfaces". In: *Computer Graphics Forum*. Vol. 17. 2. Wiley Online Library. 1998, pp. 167–174.

[Ebk+13] Hans-Christian Ebke et al. "QEx: robust quad mesh extraction". In: *ACM TOG* 32.6 (2013), p. 168.

[FM98] Kikuo Fujimura and Mihail Makarov. "Foldover-free image warping". In: *Graphical models and image processing* 60.2 (1998), pp. 100–111.

[GJC01] Guido Gerig, Matthieu Jomier, and Miranda Chakos. "Valmet: A new validation tool for assessing and improving 3D object segmentation". In: *MICCAI*. Springer. 2001, pp. 516–523.

[GKL15] Roman Getto, Arjan Kuijper, and Tatiana von Landesberger. "Extended surface distance for local evaluation of 3D medical image segmentations". English. In: *The Visual Computer* (2015), pp. 1–11. ISSN: 0178-2789. DOI: 10.1007/s00371-015-1113-z.

[Gom99] J. Gomes. *Warping and Morphing of Graphical Objects*. Computer Graphics and Geometric Modeling Series Bd. 1. Morgan Kaufmann Publishers, 1999.

[LYY08] Tong-Yee Lee, Shao-Wei Yen, and I-Cheng Yeh. "Texture Mapping with Hard Constraints Using Warping Scheme". In: *IEEE TVCG* 14.2 (2008), pp. 382–395.

[MTT13] Bogdan Mocanu, Ruxandra Tapu, and Ermina Tapu. "Mesh deformation with hard constraints". In: *Signals, Circuits and Systems (ISSCS), 2013 International Symposium on*. IEEE. 2013, pp. 1–4.

[MZX14] Y. Ma, J. Zheng, and J. Xie. "Foldover-Free Mesh Warping for Constrained Texture Mapping". In: *IEEE TVCG* 21.3 (2014), pp. 375–388.

[Tam+13] Gary KL Tam et al. "Registration of 3D point clouds and meshes: a survey from rigid to nonrigid". In: *IEEE TVCG* 19.7 (2013), pp. 1199–1217.

[Tam+14] Gary KL Tam et al. "Diffusion pruning for rapidly and robustly selecting global correspondences using local isometry." In: *ACM Trans. Graph.* 33.1 (2014), p. 4.

[VH01] Remco C Veltkamp and Michiel Hagedoorn. *State of the art in shape matching*. Springer, 2001.

[van+11] Oliver van Kaick et al. "A survey on shape correspondence". In: *Computer Graphics Forum*. Vol. 30. 6. Wiley Online Library. 2011, pp. 1681–1707.

Robust Hand Gesture Recognition from 3D Data

Vinod K Kurmi
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
vinodkk@iitk.ac.in

Garima Jain
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
gari1217@gmail.com

KS Venkatesh
Indian Institute of Technology
Dept. of Electrical Engineering
Kanpur
India (208016) , Kanpur
venkats@iitk.ac.in

Abstract

In this paper, we use the output of a 3D sensor (ex. Kinect from Microsoft) to capture depth images of humans making a set of predefined hand gestures in various body poses. Conventional approaches using Kinect data have been constrained by the limitation of the human detector middleware that requires close conformity to a standard near erect, legs apart, hands apart pose for the subject. Our approach also permits clutter and possible motion in the scene background, and to a limited extent, in the foreground as well. We make an important point in this work to emphasize that the recognition performance is considerably improved by a choice of hand gestures that accommodate the sensor's specific limitations. These sensor limitations include low resolution in x and y as well as z. Hand gestures have been chosen(designed) for easy detection by seeking to detect a fingers apart, fingertip constellation with minimum computation. without, however compromising on issues of utility or ergonomy. It is shown that these gestures can be recognised in real time irrespective of visible band illumination levels, background motion, foreground clutter, user body pose, gesturing speeds and user distance. The last is of course limited by the sensor's own range limitations. Our main contributions are the selection and design of gestures suitable for limited range, limited resolution 3D sensors and the novel method of depth slicing used to extract hand features from the background. This obviates the need for preliminary human detection and enables easy detection and highly reliable and fast (30 fps) gesture classification.

Keywords
Hand gesture recognition; Kinect; depth map; histogram; depth slicing; fingertip constellation;

1 INTRODUCTION

Human computer interaction (HCI) is not limited by physical contact with the devices. It has the potential to change the way users interact with computers and appliances, by eliminating input devices such as joysticks, mice, remote control units and keyboards, and allowing the unencumbered body to give signals to the computer through gestures. In gesture based devices, the right choice of gestures is very important for user comfort. Thus, all the gestures should be ensured to be comfortable and simple.

There are different techniques available for hand gesture recognition such as hand modeling, pattern recognition, image processing, etc. Modeling of the hand for gesture recognition has been based on Hidden Markov Model (HMM) in [eickeler1998, elmezain2008, wilson1999, fujii2014]. [suryanarayan2010] presented a 2D and 3D descriptor-based recognition system. For training the hand model, a support vector machine (SVM) was used. A state-based technique for recognition of hand gesture was also used in [bobick1997] by defining the gesture as a sequence of states. Recently, depth based gesture recognition has become quite popular in HCI. Kinect based gesture recognition systems are an example of this. [biswas2011, asad2013, raheja2011] are based on depth data provided by Kinect. Biswas [biswas2011] uses background subtraction and auto thresholding to segment the hand from an image. This allows the system to work only if hand is the first object in front of the camera. A multiclass SVM is required to train the system and individual analysis of each pixel in each frame makes it computation intensive.

Asad [asad2013], Ren [ren2013], Heickal [heickal2013] and Raheja [raheja2011] use OpenNI SDK which is dependent on predefined and available libraries like Nite middleware from Primesense, which can not be used as an open source and besides, works only when the subject is in a standing pose and facing the sensor. We thus observe many limitations in the present day literature for Kinect based hand gesture recognition.

Liu [liu2004] captures depth data using time-of-flight camera and assumes the hand to be the closest to the camera with no objects at the same depth or at a lesser depth. For their algorithm to work, the presence of the face in the frame is compulsory. Penne [penne2008] makes a similar assumption for the user to sit directly in front of the camera (distance 70-100 cm) with hand being the nearest object to the sensor.

Prisacariu [prisacariu2011] uses a combination of visual tracking and an off-the-shelf accelerometer. The tracking requires intricate 3D modeling of the hand and an accelerometer needs to be mounted on the hand, making the process cumbersome. Bigdelou [bigdelou2012] and Jaemin [jaemin2013] use built in Random Forest classifiers provided by Microsoft Kinect SDK, a closed source software, which provides skeleton modelling of human body.

In this paper, we present low complexity algorithms specifically for hand gesture recognition from the data obtained by the 3D sensor. The main goal is to detect hand gestures in varying light conditions, irrespective of neighbouring clutter, both in front of and behind the subject. In the first part, we proceed with hand segmentation with the assistance of depth histograms. Next, the fingertips of the hand are used to form a human hand specific, distinctive constellation of salient points, which yields the position and orientation of the hand and fingertips. The system is able to reject invalid gestures and works even in complete darkness, on account of exclusively depending on the depth information. Separately, we design gestures that are planned for easy detection and ergonomy, and take into consideration the sensor's specific limitations. Gesture recognition involves tracking the change in position, orientation , shape(open/close) of the hand. Gestures have also been designed for two hands used simultaneously, which keep a track of their positions with respect to each other. We have worked with a commercially available depth sensor called Kinect sold by Microsoft. The system works in real time at 30 fps. The software used for the project is Open CV in Microsoft Visual Studio using C and C++, on Windows platform with 4GB RAM. We present a brief summary of present technology and compare with our own in Table 1.

The paper is organized as follows: Section 2 explains our approach for hand segmentation from 3D sensor data and subsequent hand identification. Section 3 describes the algorithms devised for gesture recognition. The methodology adopted for distinguishing single-hand gestures and two-hand gesture is also discussed. Section 4 explains the results and experiments of the proposed algorithm. Section 5 concludes the paper with the discussion on future scope.

2 PROPOSED ALGORITHM

In our work, we have extracted depth data using Kinect sensor. The depth data sent by the Kinect is of 16 bits, with the least significant 3 bits containing index of user detected at pixel and 1 bit for error. The remaining 12 bits carry the depth information. So, the actual depth at any pixel is related to the pixel value as follows:

$$\text{Actual depth in mm} = \text{pixel value} \times \text{depth scale factor} \tag{1}$$

where depth scale factor= $\frac{2^{12}}{255}$.

2.1 Depth slicing

The essence of our approach lies in exploiting the z information of the image. A sample depth image captured from Kinect is shown in Fig. 1. We create a depth (z) histogram of the image and apply a sequence of thresholds that slice the RGBD scene into depth segments. Since we take care to design gestures that maintain a minimum hand-torso distance and a clear hand sensor distance, the hand and torso are represented as peaks in the depth histogram separated by valleys. An appropriate choice of thresholds will isolate the slice containing the hand from the rest of the scene. Fig. 2 shows the histogram and the related depth image, where the green peak in the histogram plot corresponds to the human hand and white peak to the human body. It is quite evident from the figure that the peak corresponding to the body has higher amplitude on account of the torso's much larger size.

There is a valley in the depth histogram between the hand depth peak and the human body depth peak. This valley is best suited for the depth threshold required for segmenting the hand. The first and second peaks are at d' and d'' if they satisfy (2) and (3) respectively. Subsequently, the valley is selected at depth d_v if it satisfies (4).

$$\begin{cases} h(d') > h(d) & 0 < d < d' \\ h(d') \geq h(d' - |k|) \end{cases} \tag{2}$$

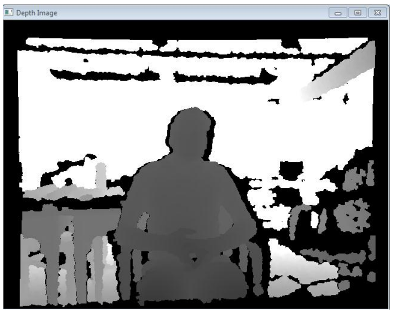


Figure 1: *Depth image*

Method	Camera Used	Foreground objects	Training Required	Real time	Additional Constraints
Biswas [biswas2011]	Kinect	Not allowed	Yes	No	No
Asad [asad2013]	Kinect	Not allowed	No	Yes	NITE Middleware Required
Raheja [raheja2011]	Kinect	Not allowed	No	Yes	NITE Middleware Required
Liu [liu2004]	TOF	Not allowed	No	Yes	Face needed in the frame
Penne [penne2008]	TOF	Not allowed	No	No	No
Ren [ren2013]	Kinect	Not allowed	No	(13.5fps)	Kinect SDK Required
Bigdelou [bigdelou2012]	Kinect	Not allowed	Yes	Yes	Kinect SDK Required
Heickal [heickal2013]	Kinect	Allowed	Yes	Yes	NITE Middleware Required
Jaemin [jaemin2013]	Kinect	Allowed	No	Yes	Kinect SDK Required
Proposed	Kinect	Allowed	No	Yes (30fps)	No

Table 1: Comparison with other reported approaches

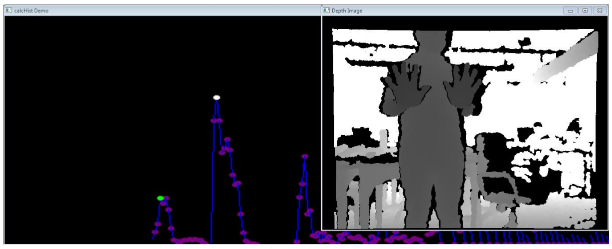


Figure 2: *Histogram and corresponding depth image*

$$\begin{cases} d'' > d' \\ h(d'') > h(d) & d' < d < d'' \\ h(d'') \geq h(d'' - |k|) \end{cases} \tag{3}$$

$$\begin{cases} d'' > d_v \\ d' < d_v \\ h(d_v) < h(d') \\ h(d_v) < h(d'') \\ h(d_v) \leq h(d” - |k|) \end{cases} \tag{4}$$

where $|k|$ <10 cm is any small integer and $h(d)$ is the histogram function at any depth value d.

Next, we segment the hand using any threshold $t \leq v - d$ into the binary hand indicator image $h(x,y)$ using (5).

$$h(x,y) = 1, \text{iff } d - t < z < d + t \tag{5}$$

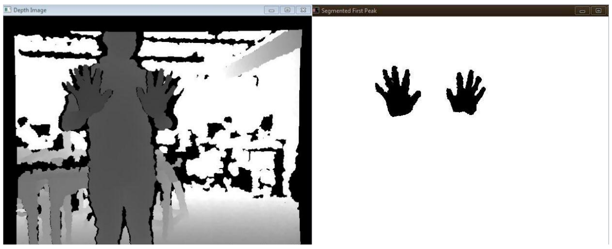


Figure 3: *Hand segmentation*

In Fig. 3, the segmented region of hand is shown in black and the background pixels are shown in white. This segmentation output is a result of depth thresholding to isolate the first peak.

In Fig. 6, both hands present the gesture at two different depths, and the above method is extended to isolate each hand through a sequence of depth slices. Also, if some object precedes the slices containing the hands, then the foremost slice is rejected and detection is attempted in the subsequent slices.

2.2 Fingertip constellation and hand orientation

Fingertip constellation of the hand is the basis for identification of hand, since, in the open hand, fingertips are always arranged in their unique configuration in space. There may be some other objects which slightly resemble this configuration, but the human hand yet has many distinctive properties that distinguish it from other objects.

In the segmented hand binary image, the contour is detected, and is enclosed in its convex hull. High curvature points on the hull represent fingertips. Fingertip positions as well as the maximum convexity defect points between consecutive fingertips are both noted as feature points. They together constitute the constellation of points representing the hand.

2.2.1 Geometrical Properties of Hand

There is a possibility of objects being present at the same depth as the hand, and these objects would not be eliminated by depth slicing alone. These are filtered out, first, with a size filter (hand area is expected to be 1000 to 8000 pixels). A second shape filter examines the feature point constellation of the object.

Suppose that array **P** contains convex hull points

P= $[p_1, p_2, ...]$

and corresponding convexity defects points are stored in the array **Q**

Q= $[q_1, q_2, ...]$

The points p_i and q_i form a vector $\bar{r}_i$, and p_{i+1} and q_i form a vector $\bar{s}_i$. We focus on the angle between vector $\bar{r}_i$ and vector $\bar{s}_i$.

$$\cos(\theta) = \frac{\bar{r}_i.\bar{s}_i}{||\bar{r}_i||.|\bar{s}_i||} \qquad (6)$$

This angle is always less than 60° (only between thumb and index finger it goes to 90°), for the real fingertip. This criterion removes all the false fingertip points.

2.2.2 *Orientation of Hand*

To detect hand orientation, we determine s_m, the centre of the palm as the centre of the largest circle inscribable within the palm and s_c, the centre of the convex hull of the hand. With our choice of gestures, we ensure that the center point of convex hull s_c (red dot inside the maximally inscribed circle in Fig. 8) is always different from the palm center point s_m (blue dot inside the palm in Fig. 8), point s_c always lies towards the fingers as compared to the point s_m. The direction vector $\vec{p}$ from s_m to s_c measures the hand orientation.

2.2.3 *Left and Right Hand*

For hand disambiguation, one uses the differences between the two hand constellations in relation to the current hand orientation. When the orientation is vertical, left and right hands can be disambiguated by the difference in the thumb position. When the hand is horizontal, then we look at the location of the hand constellation with respect to s_c to disambiguate the hands.

2.3 Two hand gestures

In the two-hand case, the first peak of the depth histogram corresponds to both the hands if both hands are at the same depth. After segmenting the hand with the help of depth data, we analyse all the contours in the depth slice. If two contours satisfy the hand identification criteria, we conclude that two hands are present. But, if both the hands are at different depths, they occur in different depth slices. We avoid this condition for two hand gesture case and treat this as a single hand gesture. The hand nearer to the sensor is considered for gesture analysis.

3 PROPOSED GESTURE IDENTIFICATION ALGORITHM

In the previous section, we have explained the isolation of hand from a scene. This section deals with the gesture identification algorithms. The first step is to decipher if the gesture is single or two handed. Then the hand position is tracked from the reference, and the system identifies the gesture performed by the user, if valid. If an invalid gesture is performed, the system rejects it.

3.1 Detection of the number of hands

Fig. 4 outlines the steps used to detect if the gesture is performed by one hand or both the hands. This algorithm also eliminates the foreground objects.

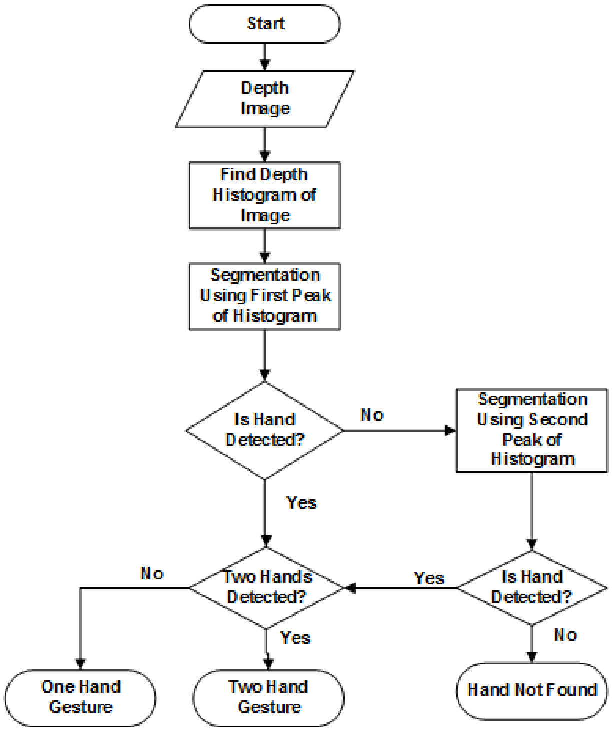


Figure 4: *Flowchart for the detection of number of hands*

3.2 Identification of the gesture

In a gesture recognition system, only valid gestures, i.e, gestures predefined in our library, should be identified and invalid gestures should be ignored. In our algorithm, before performing a gesture, the user has to set a reference point for that gesture. This allows for device initialization, the duration of which maybe set as an operational parameter. The hand center point, if maintained steady for the chosen initialization time, is accepted as the reference point for the rest of the gesture. All the gestures described by us are defined by the relative position of the hand reference point with respect to the palm center point s_m.

All gestures also end with a predefined closing pose called the “gesture over" state after which the reference point is cleared. A reference point has to be created again for performing the next gesture. The presence of the initialisation step ensures that inadvertent hand movements are not considered for processing.

Different gestures for single hand and two hands are described below with the algorithms listed for their identification. This gesture vocabulary is scalable in terms of the number of gestures, using various combinations of both the hands in different orientations.

3.2.1 *Single hand gestures*

There are six gestures designed for single hand implementation and are listed in the Table 2.

The algorithm to identify a single hand gesture, i.e., Algorithm 1, starts when it has been identified that a single hand is going to perform the gestures.

Algorithm 1 Identification of single hand gestures

```
start:
sm ← center of palm
i ← threshold for reference point creation
j ← threshold for change in area
k ← threshold for decrease in depth
l ← threshold for rate of decrease in depth
first:
if reference point exists then
    goto top2
else
    goto top1
top1:
if sm constant for time > i then
    create reference point
else
    goto start
top2:
if change in area > j then
    if count==1 then
        gesture ← mute
    else
        gesture ← unmute
    count ← !count
    goto last
top3:
if decrease in depth > k then
    if rate of decrease in depth > l then
        gesture ← OFF
    else
        gesture ← Invalid Gesture
    goto last
top4:
if hand is moving then
    if hand orientation==left then
        goto top5
    else
        goto top6
else
    goto last
top5:
if hand is moving up then
    gesture ← Channel Up
else
    gesture ← Channel Down
goto last
top6:
if hand is moving up then
    gesture ← Volume Up
else
    gesture ← Volume Down
goto last
last:
Gesture over
goto start
```

S.N.	Hand Gesture	Meaning
1	Horizontal right hand moved upward	Channel Up
2	Horizontal right hand moved downward	Channel Down
3	Horizontal left hand moved upward	Volume Up
4	Horizontal left hand moved downward	Volume Down
5	Vertical hand (L/R) closed	Mute/Unmute
6	Vertical hand (L/R) moved towards the sensor	OFF

Table 2: Single hand gestures

A few of the gestures are specific to whether the gesture is performed by left or right hand. But some gestures are independent to hand identification, like Mute/Unmute and OFF. These gestures can be done with either of the hands.

3.2.2 Two hand gestures

We have defined ten two-hand gestures for this work, with four being static and six being dynamic gestures, as in Fig. 5. In the dynamic gestures, at least one of the hands is in motion. It should however be kept in mind that the two hands do not overlap while performing the gesture. They should be kept at a sufficient distance from each other. These gestures are using Algorithm 2.

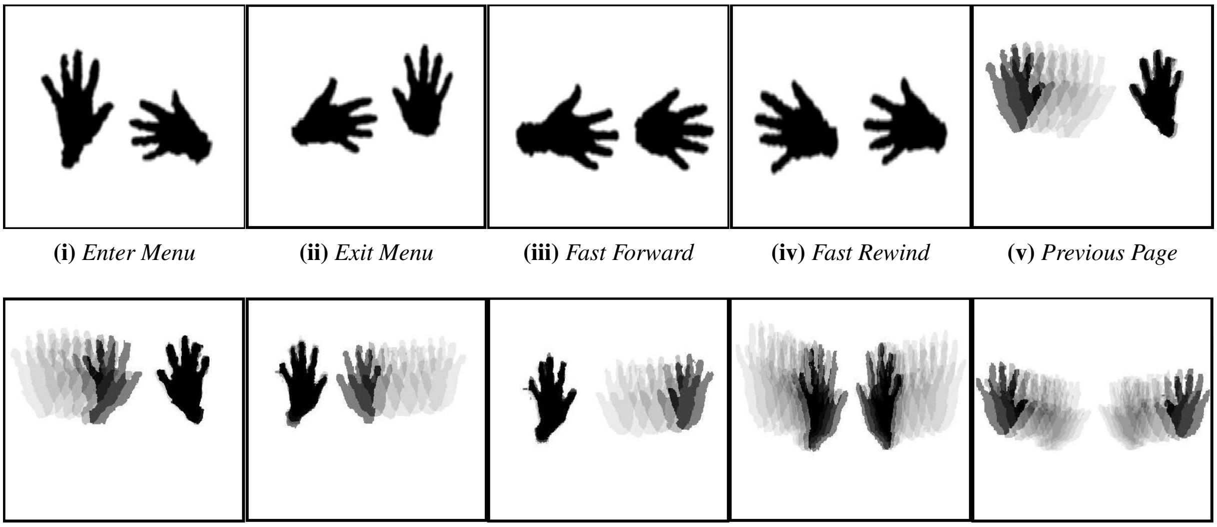
4 RESULTS AND EXPERIMENTS

Initially, the hand is segmented from the scene with the aid of depth histogram. The contour detection of the segmented depth slice is shown in Fig. 7. Fig. 8 shows the convex hull points and convexity defect points of the contour. The purple dots are the convexity defect maxima and red dots are the fingertips. A maximal inscribed circle is drawn inside the hand.

Depth based hand segmentation depends upon the peak value of depth histogram. It is possible that this peak does not correspond to hand or there might be other objects present at the same depth as hand. The fingertip constellation detection used by us can deal with these situations comfortably. Some of the cases are presented here.

4.1 Spurious object present at same depth

The objects which are present at the same depth level of the hand, do not get eliminated after depth slicing. The convex hull points and convexity defects points configuration for them cannot be the same as of the hand, and thus allows us to eliminate such objects.



(i) *Enter Menu* **(ii)** *Exit Menu* **(iii)** *Fast Forward* **(iv)** *Fast Rewind* **(v)** *Previous Page*

(vi) *Next Page* **(vii)** *Scroll Left* **(viii)** *Scroll Right* **(ix)** *Zoom In* **(x)** *Zoom out*

Figure 5: *Two Hand Gestures*

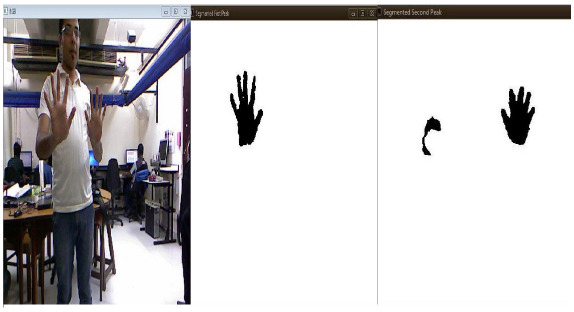


Figure 6: *Segmentation using first and second peak*

4.2 Spurious objects in foreground

In the gesture recognition environment, it is also possible that the first peak of histogram does not satisfy the properties of a human hand. This means the hand is not the nearest object from the depth sensor. In this case, we have to look at the next depth slice. This process ideally can be continued until the human hand is found. But there are also limitations of range as well as view angle of the camera. The foreground objects should not be too large, otherwise they will cover most of the image.

Fig. 9 shows the segmentation using the second peak of the depth histogram. In this figure, we have segmented the second object from the sensor or in other words it is the second depth slice. Based on the fingertip constellation, the human hand is identified and remaining objects are ignored. The figure shows that only the hand is detected, even in the presence of other objects in the foreground or at the same depth.

4.3 Recognition Performance Analysis

The performance of the recognition system is only slightly affected to an extent by careless gesturing. This

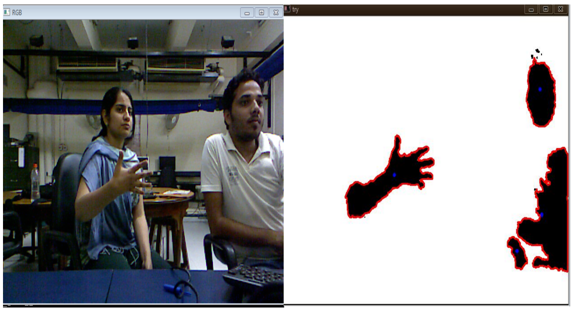


Figure 7: *Contour of the hand*

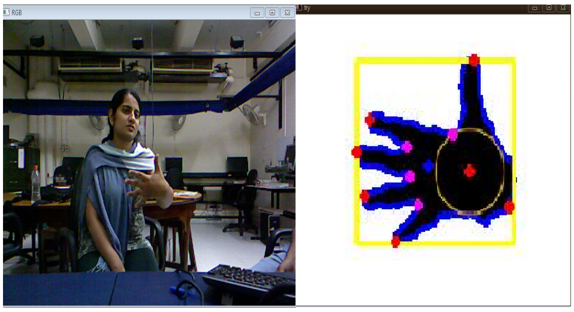


Figure 8: *Convex hull points and convexity defects points with palm center*

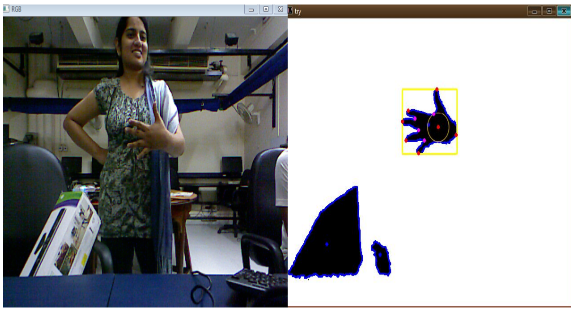


Figure 9: *Identification of the hand in the presence of other objects*

Algorithm 2 Identification of two hand gestures

```
start:
sm ← center of palm
i ← threshold for reference point creation
first:
if reference point exists then goto top2
else goto top1
top1:
if sm constant for time > i then
    create reference point
else goto start
top2:
if both hands vertical then goto top3
else
    if both hands horizontal then goto top7
    else goto top8
top3:
if both hands moving then
    goto top4
else
    if only left hand moving then
        goto top5
    else goto top6
top4:
if both hands moving outward then
    gesture ← Zoom Out
else
    gesture ← Zoom In
goto last
top5:
if hand moving outward then
    gesture ← Next Page
else
    gesture ← Previous Page
goto last
top6:
if hand moving outward then
    gesture ← Scroll Right
else
    gesture ← Scroll Left
goto last
top7:
if both hands oriented towards left then
    gesture ← Fast Forward
else
    gesture ← Fast Rewind
goto last
top8:
if left hand horizontal then
    gesture ← Enter Menu
else
    gesture ← Exit Menu
goto last
last:
Gesture over
goto start
```

is not to deny that the recognizer has considerable, and tunable recognition acceptance tolerance incorporated in its design. Yet, a gesture performed in an extremely wrong way, will result in lesser accuracy as compared to our sample set results. All the gestures work fine in real time scenarios. Our gesture vocabulary is intuitive and comfortable. Sensitivity to body pose variation and background and foreground clutter is minimal. Experiments to evaluate the performance of the gesture recognition system have been conducted, with each gesture performed 3 times by 15 individuals. The demonstration videos clearly reflect the robustness and accuracy of the system, irrespective of neighbouring clutter, user body pose, and its completely invariance to visible band illumination in indoor conditions.

Gesture	Correct Recognition	Unsuccessful Recognition
Channel up	45/45	0/45
Channel down	45/45	0/45
Volume up	45/45	0/45
Volume down	45/45	0/45
Mute/Unmute	45/45	0/45
OFF	45/45	0/45

Table 3: Single hand gestures

Gesture	Correct Recognition	Unsuccessful Recognition
Enter Menu	45/45	0/45
Exit Menu	45/45	0/45
Fast Forward	44/45	1/45
Fast Rewind	44/45	1/45

Table 4: Two hand static gestures

Gesture	Correct Recognition	Unsuccessful Recognition
Zoom In	45/45	0/45
Zoom Out	45/45	0/45
Next Page	45/45	0/45
Previous Page	45/45	0/45
Scroll Left	45/45	0/45
Scroll Right	45/45	0/45

Table 5: Two hand dynamic gestures

5 CONCLUSIONS

The ability of our algorithm to work without any intricate modelling of the hand or the use of heavy machine learning techniques, enables it to work straight away without any sort of user dependent training whatsoever. We develop a novel approach for segmenting the hand, based on the constellation of fingertip points in space, while ensuring that the design of gestures (all five fingertips of the hand are visible to the sensor with a minimal hand-torso distance) accommodates the limitations of the sensor.

This approach is robust and accurate irrespective of neighbouring clutter, varying lighting conditions, user body pose, and is completely invariant to visible band illumination in indoor conditions. To our knowledge, no currently available system possesses all these features.

The system working range is from 0.6 m to 2.5 m. The working range of the system can be improved when sensors that operate over a wider range are available. The prototype can be further extended for multi-user control, finger gesture recognition and to distinguish between palm and the back of the hand using a higher resolution depth camera.

6 ACKNOWLEDGMENTS

This work was supported by Samsung India Electronics Private Limited as a part of a research project.

7 REFERENCES

[asad2013] Asad, M.; Abhayaratne, C., "Kinect depth stream pre-processing for hand gesture recognition," Image Processing (ICIP), 2013 20th IEEE International Conference on , vol., no., pp.3735,3739, 15-18 Sept. 2013. doi: 10.1109/ICIP.2013.6738770.

[bigdelou2012] Bigdelou, Ali, Tobias Benz, Loren Schwarz, and Nassir Navab. "Simultaneous categorical and spatio-temporal 3d gestures using kinect." In 3D User Interfaces (3DUI), 2012 IEEE Symposium on, pp. 53-60. IEEE, 2012.

[biswas2011] Biswas, K. K.; Basu, S.K., "Gesture recognition using Microsoft KinectÂ®," Automation, Robotics and Applications (ICARA), 2011 5th International Conference on , vol., no., pp.100,103, 6-8 Dec. 2011 doi: 10.1109/ICARA.2011.6144864.

[bobick1997] Bobick, Aaron F and Wilson, Andrew D, "A state-based approach to the representation and recognition of gesture, " Pattern Analysis and Machine Intelligence, IEEE Transactions no 12, 1997 vol. no. 19, pp. 1325-1337.

[eickeler1998] Eickeler, S.; Kosmala, A; Rigoll, G., "Hidden Markov model based continuous online gesture recognition," Pattern Recognition, 1998. Proceedings. Fourteenth International Conf. on , pp.1206,1208 vol.2, 16-20 Aug 1998.

[elmezain2008] Elmezain, M.; Al-Hamadi, A; Appenrodt, J.; Michaelis, B., "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , vol., no., pp.1,4, 8-11 Dec. 2008.

[fujii2014] Fujii, Tatsuya, Jae Hoon Lee, and Shingo Okamoto. "Gesture Recognition System for Human-Robot Interaction and Its Application to Robotic Service Task." In Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists. 2014.

[heickal2013] Heickal, Hasnain, Tao Zhang, and Md Hasanuzzaman. "Real-time 3D full body motion gesture recognition." In Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on, pp. 798-803. IEEE, 2013.

[jaemin2013] Jaemin, Lee, H. Takimoto, H. Yamauchi, A. Kanazawa, and Y. Mitsukura. "A robust gesture recognition based on depth data." In Frontiers of Computer Vision, 2013 19th Korea-Japan Joint Workshop on, pp. 127-132. IEEE, 2013.

[penne2008] Penne, Jochen, Stefan Soutschek, Lukas Fedorowicz, and Joachim Hornegger. "Robust real-time 3D time-of-flight based gesture navigation." In Automatic Face and Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, pp. 1-2. IEEE, 2008.

[raheja2011] Raheja, J.L.; Chaudhary, A; Singal, K., "Tracking of Fingertips and Centers of Palm Using KINECT," Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on , vol., no., pp.248, 252, 20-22 Sept. 2011.

[ren2013] Ren, Zhou, J. Yuan, J. Meng, and Z. Zhang. "Robust part-based hand gesture recognition using kinect sensor." Multimedia, IEEE Transactions on 15, no. 5 (2013): 1110-1120.

[suryanarayan2010] Suryanarayan, P.; Subramanian, A; Mandalapu, D., "Dynamic Hand Pose Recognition Using Depth Data," Pattern Recognition (ICPR), 2010 20th International Conference on , vol., no., pp.3105,3108, 23-26 Aug. 2010.

[wilson1999] Wilson, Andrew D and Bobick, Aaron F, "Parametric hidden markov models for gesture recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions 1999, vol.no 21, pp.884-900.

[liu2004] Liu, Xia, and Kikuo Fujimura. "Hand gesture recognition using depth data." In Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pp. 529-534. IEEE, 2004.

[prisacariu2011] Prisacariu, Victor Adrian, and Ian Reid. "Robust 3D hand tracking for human computer interaction." In Automatic Face and Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 368-375. IEEE, 2011.

Inpainted image quality assessment based on machine learning

V. Voronin[1] V. Marchuk[1] E. Semenishchev[1] S. Maslennikov[1] I. Svirin[2]

[1]Don State Technical University
Shevchenko 147
346500, Shakhty, Russian Federation
voronin_sl@mail.ru

[2]CJSC Nordavind
Varshavskoe 125
Moscow, Russian Federation
head@nordavind.ru

ABSTRACT

In many cases inpainting methods introduce a blur in sharp transitions in image and image contours in the recovery of large areas with missing pixels and often fail to recover curvy boundary edges. Quantitative metrics of inpainting results currently do not exist and researchers use human comparisons to evaluate their methodologies and techniques. Most objective quality assessment methods rely on a reference image, which is often not available in inpainting applications. This paper focuses on a machine learning approach for no-reference visual quality assessment for image inpainting. Our method is based on observation that Local Binary Patterns well describe local structural information of the image. We use a support vector regression learned on human observer images to predict the perceived quality of inpainted images. We demonstrate how our predicted quality value correlates with qualitative opinion in a human observer study.

Keywords

Inpainting, quality assessment, metric, visual salience, machine learning.

1. INTRODUCTION

Objective image quality metrics are designed to predict perception by humans based on an image processing without a human observer being involved. Such metric allow assessing an image quality quickly, but existing metrics behave differently in comparison with a quality perceived by human observers. Most of existing methods implement full-reference metrics where complete reference image is assumed to be known. In the case of image inpainting reference image just does not exist. This situation requires a no-reference or "blind" quality assessment approach.

Objective methods for assessing perceptual image quality have traditionally attempted to quantify the visibility of errors between a distorted image and a reference image using a variety of known properties of the human visual system. The most fundamental problem with the traditional approach is the definition of image quality. In particular, it is not clear that artifact visibility should be associated with loss of quality. Some artifacts may be clearly visible but very hard to model numerically.

Several works on objective image inpainting quality assessment have been published in recent years. For instance, an analysis of gaze patterns was involved to quality assessment in work [Ven10]. Authors postulate that perceived by human image quality is related to so-called “saliency”. To quantitatively assess saliency, they compute the gaze density for a given image inside and outside the inpainted region. Resulting quality estimates are achieved as a relation of the gaze densities of an image inside and outside the hole region. Authors have used an eye tracker to estimate a gaze density. This method has the same disadvantages as subjective evaluation.

Most of proposed approaches use saliency maps to estimate visibility of different artifacts in inpainted region. The key idea is based on the change of the saliency map before and after inpainting. In paper [Pau09] this problem addressed by two proposed metrics: average squared visual salience (ASVS) and degree of noticeability (DN). Drawbacks of these metrics are related to the fact that they do not take into consideration the global visual appearance of the image. In [Pau09] proposed another visual saliency based metric. He defined a normalized gaze density measure that uses the original image as a reference, and shows that if there is any change in the saliency map corresponding to the inpainted image, then this change is related to the perceptual quality of the inpainted image. Authors use the visual coherence of the recovered regions and the visual saliency describing the visual importance of an area. This approach shows promising results but addresses only few possible inpainting artifacts.

There is a work that generalize some previous methods like Structural Similarity Index (SSIM)

[Pau10] for image inpainting. This approach is able to achieve good results, but it's completely lacking a high level modeling of a human visual system.

At this point, we may conclude that abovementioned approaches are quite efficient for particular tasks. Nevertheless, existing approaches are weakly correlated with a human perception and, thus, additional investigation on this topic is needed.

2. IMAGE INPAINTING QUALITY

At first the inpainting problem was approached as "error concealment" in the field of telecommunications. The goal of this technique was to fill-in image blocks that have been lost during data transmission. More recently, more elaborated techniques for digital image inpainting such as one presented by Bertalmio et al. [Ber01] have been developed. During the last decade, many methods addressing inpainting problem have been proposed. It leads to the natural need of robust inpainting performance metric. Typically, subjective expert-based approaches are involved which is expensive and time consuming procedure. So, alternative approach has to be developed to address the problem of objective image quality assessment.

The problem of inpainted image quality assessment is highly related to the human visual system modeling problem. In order to design a good quality metric one should take into account its different properties. One way to do this is to model it with machine learning techniques.

Let's introduce basic notations used in our work. The whole image domain I is composed of two disjoint regions: the inpainting region Ω, and the known region Φ $(\Phi = I - \Omega)$ as shown at the figure 1.

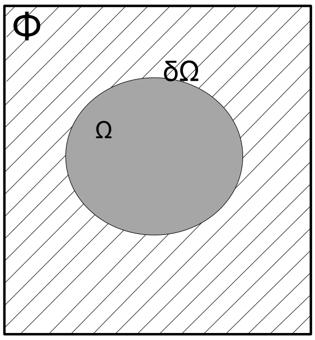


Figure 1. Image model.

Given an image I and a region Ω inside it, the inpainting problem consists in modifying image values of the pixels in Ω so that this region does not stand out with respect to its surroundings. The purpose of inpainting might be to restore damaged portions of an image (e.g. an old photograph where folds and scratches have left image gaps) or to remove unwanted elements present in the image (e.g. a microphone appearing in a film frame). The region Ω is always given by the user, thus the localization of Ω is not a part of the inpainting problem.

It is very difficult to compare the "original" image and an inpainted one, because inpainted region can be large and very different from the corresponding region of the original image. In some cases, a visual image quality may be nearly perfect, but objective quality in terms of pixel-oriented metrics like PSNR will be poor. On way to model human attention and to estimate the visibility of different image areas is to use so-called saliency maps. We exploit this approach together with machine learning to model relations between local geometric patterns and perceived by a human observer image quality.

3. THE PROPOSED METHOD

In [Fra14], we have proposed the inpainting quality assessment technique based on a machine learning approach. Our method allows to receive both low and high level inpainted image descriptions. Next, we have used a support vector regression learned on human observer images to predict the perceived quality of inpainted images. One of the major problems there was a computational complexity.

The main workflow of proposed method is presented on the Figure 2.

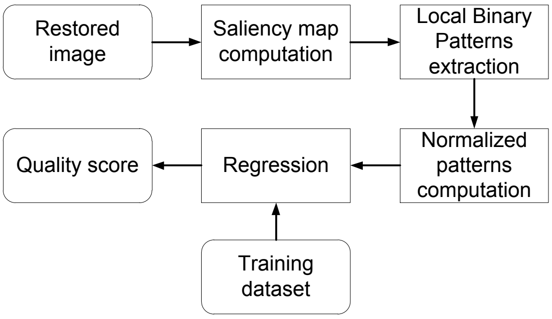


Figure 2. Overall algorithm block scheme.

The first step is to compute an importance of each sub-region of inpainted area. To approach this problem, we use a visual saliency which plays an important role in human visual perception. Human eye at each time clearly sees only a small portion of the space, while a much larger portion of the space is perceived very 'blurry'. The latest information is sufficient to assess the importance of different areas and to draw attention to important areas of a visual field. Most of methods give so-called saliency map: a two-dimensional image in which each pixel value is related to an importance of this region.

It is believed that two stages of visual processing are involved: first, the parallel, fast, but simple pre-attentive process; and then, the serial, slow, but complex attention process. Innovation denotes the novelty part, and prior knowledge is the redundant information that should be suppressed. In the field of image statistics, such redundancies correspond to statistical invariant properties of our environment. It is widely accepted that natural images are not random, they obey highly predictable distributions. In the following sections, we will demonstrate a method to approximate the "innovation" part of an image by removing the statistical redundant components. This part, we believe, is inherently responsible to the popping up of regions of interest in the pre-attentive stage.

In our work, we use spectral residual approach [Xia07]. It defines an entropy of the image as:

$$H(Image) = H(Innovation) + H(Prior\ Knowledge)\ .$$

This model is independent of features, categories, or other forms of prior knowledge of the objects. By analyzing the log-spectrum of an input image, authors extract the spectral residual of an image in spectral domain. They have proposed a fast method to construct the corresponding saliency map in spatial domain.

Given an input image I with a Fourier decomposition F, the log spectrum $L(F)$ is computed from the down-sampled image with height equal to 64 pixels.

If the information contained in the $L(F)$ is obtained previously, the information required to be processed is:

$$H(R(F)) = H \cdot L(F)|A(F),$$

where $A(F)$ denotes the general shape of log spectra, which is given as a prior information. $R(F)$ denotes the statistical singularities that is particular to the input image. To compute area importance metric we have used the following expression:

$$Q = \frac{1}{\|\Phi\|} \cdot \sum_{p \in \Phi} S(p)\,,$$

where S is the saliency map corresponding to the inpainted image, which gives $S(p)$ as the saliency map value corresponding to pixel p. We have used Q value as a threshold level at the next step and calculated the assessment only for those recovered areas for which $S(p) > Q$.

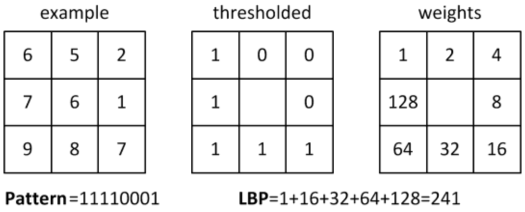
The saliency map is an explicit representation of proto-objects [Tan11]. We use a simple threshold segmentation to detect proto-objects in a saliency. Given $S(x)$ of an image, the object map $O(x)$ is obtained:

$$O(x) = \begin{cases} 1 & if\ S(x) > threshold \\ 0 & otherwise \end{cases}\ .$$

Empirically, we set $threshold = E(S(x)) \times 3$, where $E(S(x))$ is the average intensity of the saliency map. While the object map $O(x)$ is generated, proto-objects can be easily extracted from their corresponding positions in input image.

After that we perform feature extraction for found proto-objects. Features are characteristic properties of the artifacts whose value should be similar for artifacts in a particular class, and different from the values for artifacts in another. The choice of appropriate features depends on the particular application.

At present work we use local binary pattern operator (LBP), introduced by Ojala et al. [Tim02]. It is based on the assumption that texture has locally two complementary aspects, a pattern and its strength. The LBP was proposed as a two-level version of the texture unit to describe the local textural patterns. As the neighborhood consists of 8 pixels, a total of 256 different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood. An example of an LBP computation is shown on Figure 3.


example

6	5	2
7	6	1
9	8	7

thresholded

1	0	0
1		0
1	1	1

weights

1	2	4
128		8
64	32	16

Pattern=11110001 **LBP**=1+16+32+64+128=241

Figure 3. Example of local binary pattern computation.

Learning stage involves word frequency histogram of local binary patterns in salient regions as a feature vector and Support Vector Regression (SVR) as a learning method. Illustration of feature vector is presented in Figure 4.

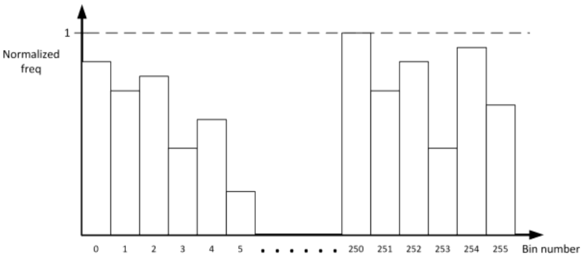


Figure 4. Example of normalized pattern histogram.

Support Vector Machine (SVM) and other kernel methods have achieved a lot of attention recent years, and it has been reported to outperform nearest neighbor classifier in texture classification. Also, boosting based approaches such as AdaBoost, and bagging classifiers like the Random Forest classifier have been successfully applied to texture classification. The problem of texture retrieval in some extent is related to texture classification.

To compare histograms we use Earth Movers Distance (EMD) [Lev01]. This is a common way to compare two probability distributions (in our case presented by histograms). To incorporate EMD distance into the SVM framework, we use extended Gaussian kernels:

$$K(S_i, S_j) = \exp(-\frac{1}{A} D(S_i, S_j)) ,$$

where $D(S_i, S_j)$ is EMD if S_i and S_j are image signatures. The resulting kernel is the EMD kernel, A is a scaling parameter that can be determined through cross-validation. We have found, however, that setting its value to the mean value of the EMD distances between all training images gives comparable results and reduces the computational cost.

To learn a regression function, we use a support vector machine regression. As a result, the classification algorithm can be written as:

$$a(x) = sign\left(\sum_{i=1}^{n} \lambda_i c_i x_i \cdot x - b\right).$$

We will use $a(x)$ as a predicted value of image quality.

4. EXPERIMENTAL RESULTS

The experimental method for the subjective quality assessment was chosen the Mean Opinion Score (MOS) [Rib11]. The MOS values are based on subjective data obtained from the experiment. Participants were presented with one inpainted image at a time in a random order and different to each observer. Given an image, the participants were asked to judge the overall image quality of the inpainted image using the quality scale: Excellent, Good, Fair, Poor, Bad. In order to be able to analyse the obtained subjective data, each of the five adjectives in the descriptive quality scale had an equivalent numerical value, or score (not shown to the observers). Accordingly, Excellent corresponded to a 5 score and Poor to a 1 score. The MOS was obtained for each reproduction by computing the arithmetic mean of the individual scores given by participants:

$$MOS = \frac{1}{n} \sum_{i=1}^{n} \mathrm{Score}_i ,$$

where n denotes the number of observers, and Score_i the score given by the observer to the inpainted image under consideration. The criterions used to estimate quality presented in the Table 1.

MOS	Quality	Criteria
5	Excellent	Artifacts are Imperceptible
4	Good	Artifacts are perceptible buy not annoying
3	Fair	Artifacts are slightly annoying
2	Poor	Artifacts are annoying
1	Bad	Artifacts are very annoying

Table 1. Quality criteria's

For evaluation purposes we use database of 300 images. Note, that the test images have been chosen to have different geometrical features: texture, structure and real images. After applying the missing mask, all images have been inpainted by four different methods [Oli01, Ber00, Tel04, Cri04]. For each inpainted image, its quality was assessed by 10 human observers. The results were divided into two disjoint subsets. The first was used for training, the second - to verify the results. Some of images from test database are presented at Figures 5-7 (a - images with missing pixels, b - images reconstructed by the Smoothing, c - images reconstructed by the Navier-Stokes, d - images reconstructed by the Telea, e - images reconstructed by the EBM).

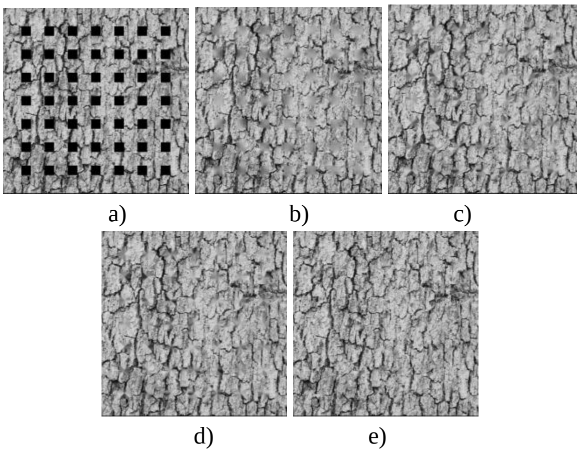


Figure 5. Examples of texture images from test database.

Thus, in an attempt to establish a ranking of the considered algorithms in terms of perceived quality of the inpainted images, and considered the database described above, a psychophysical experiment will be carried out, according to the specifications. The obtained raw perceptual data will be statistically analyzed in order to determine the ranking of the inpainting algorithms. To evaluate the objective quality assessment methods, we use the MOS. Furthermore, the prediction accuracy of proposed metric was evaluated using Spearman rank order correlation coefficient (SRCC) for proposed metric results and subjective MOS estimation. Results of numeric comparison are presented in the Table 2.

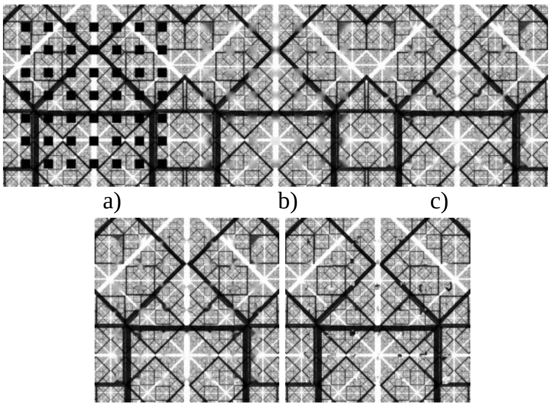


Figure 6. Examples of structure images from test database.

Proposed objective quality metric shows strong correlation with perceived by human quality. Thus, our approach is quite efficient to estimate quality of the inpainted images.

Methods		*MOS*	$\overline{MOS}$	*SRCC*
Smoothing [Oli01]	texture	2.21	2.08	0.84
	structure	1.58		
	image	**2.45**		
Navier-Stokes [Ber00]	texture	**3.15**	2.69	0.95
	structure	1.73		
	image	3.21		
Telea [Tel04]	texture	3.08	2.78	**0.96**
	structure	2.02		
	image	**3.23**		
EBM [Cri04]	texture	**4.41**	**3.69**	0.93
	structure	3.13		
	image	3.54		

Table 2. Spearman rank correlation of proposed metric results and subjective MOS estimation

Results of comparison Spearman rank order correlation coefficient, which finds the linear relationship between two variables using the formula for our method with several popular methods are presented in Tables 3.

DN	**SSIM**	**ASVS**	**PROPOSED METRIC**
0.61	0.71	0.68	**0.78**

Table 3. CC comparison

These tables show that our approach outperforms known and widely used algorithms on a selected image dataset in term of correlation coefficient.

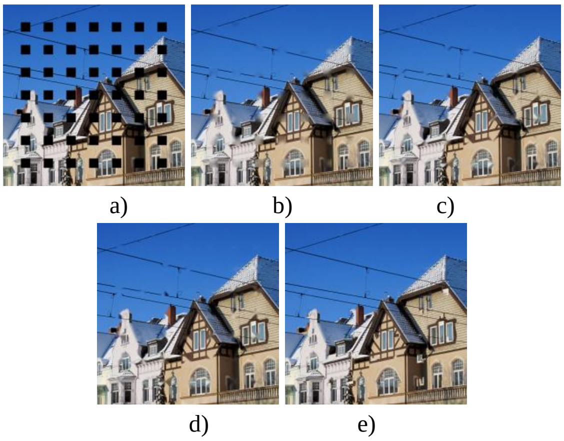


Figure 7. Examples of real images from test database.

5. CONCLUSION

In this work we have presented a novel no-reference inpainting quality assessment technique which is based on a machine learning approach. Our method use inpainted image description by local binary patterns weighted by visual importance. Next, we have used a support vector regression learned on human observer images to predict the perceived quality of inpainted images. We have demonstrated that predicted quality value highly correlates with a qualitative opinion in a human observer study.

6. ACKNOWLEDGMENTS

The reported study was supported by the Russian Foundation for Basic research (RFBR), research project №15-37-21124\15 and grant of the President of the Russian Federation for the young scientists №MK-6986.2015.8.

7. REFERENCES

[Ber01] Bertalmio, M., Bertozzi, A., Sapiro, G. Navier-Stokes, fluid dynamics, and image and video inpainting, Hawaii: Proc. IEEE Computer Vision and Pattern Recognition (CVPR), pp. 213-226, 2001.

[Ber00] Bertalmio, M., Sapiro, G., Caselles, V. and Balleste, C. Image inpainting, New Orleans: Proceedings of SIGGRAPH, pp. 102-133, 2000.

[Cri04] Criminisi, A., Perez, P., and Toyama K. Region filling and object removal by exemplar-based image inpainting, IEEE Transactions on Image Processing 13, pp. 1200–1212, 2004.

[Fra14] Frantc, V.A., Voronin, V.V., Marchuk, V.I., Sherstobitov, A.I., Agaian, S., Egiazarian, K. Machine learning approach for objective inpainting quality assessment, Proc. SPIE 9120, Mobile Multimedia/Image Processing, Security, and Applications 2014, 91200S, 2014.

[Lev01] Levina, E., Bickel, P. The Earth Mover's Distance is the Mallows Distance: Some Insights from Statistics, Proceedings of ICCV 2001 (Vancouver, Canada), pp. 251–256, 2001.

[Oli01] Oliveira, M., Bowen, B., Kenna, R. Mc, and Chang, Y.-S. Fast Digital Image Inpainting, In Proc. VIIP, pp. 261-266, 2001.

[Pau09] Paul A., and Singhal, A. Visual salience metrics for image inpainting, Proc. IS&T/SPIE Electronic Imaging, 2009.

[Pau10] Paul, A., Singhal, A., and. Brown, C. Inpainting quality assessment, Journal of Electronic Imaging, vol. 19, pp. 011002-011002, 2010.

[Rib11] Ribeiro, F., Florencio, D., Cha, Zhang, Seltzer, M. CROWDMOS: An approach for crowdsourcing mean opinion score studies, IEEE International Conference on, Acoustics, Speech and Signal Processing (ICASSP), pp. 2416 – 2419, 2011.

[Tan11] Tang, Huixuan, Neel, Joshi, and Ashish, Kapoor. Learning a blind measure of perceptual image quality, IEEE Conference on, Computer Vision and Pattern Recognition (CVPR), pp. 305 – 312, 2011.

[Tel04] Telea, A. An image inpainting technique based on the fast marching method, Journal of Graphics Tools, vol. 9, no. 1, ACM Press, pp. 25-36, 2004.

[Tim02] Timo, Ojala, Pietikainen, Matti, and Maenpaa, Topi. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on, Pattern Analysis and Machine Intelligence, pp. 971-987, 2002.

[Ven10] Venkatesh, Vijay, M., and Cheung, S.S. Eye tracking based perceptual image inpainting quality analysis, Image Processing (ICIP), 17th IEEE International Conference on IEEE, pp. 1109 – 1112, 2010.

[Xia07] Xiaodi, Hou, and Zhang, Liqing. Saliency detection: A spectral residual approach, IEEE Conference on, Computer Vision and Pattern Recognition, 2007.