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

CSRN 2803

(Eds.)

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

Computer Science Research Notes

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

Proceedings

WSCG 2018

Posters Proceedings

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

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

CSRN 2803

(Eds.)

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

Computer Science Research Notes

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

Proceedings

WSCG 2018

Posters Proceedings

Vaclav Skala - UNION Agency

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

Computer Science Research Notes
CSRN 2803

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

Managing Editor: Vaclav Skala

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

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

ISSN 2464-4617 (Print) ***ISSN 2464-4625*** (CD/DVD)
ISBN 978-80-86943-42-8 (CD/-ROM)

WSCG 2018

International Program Committee

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

WSCG 2018

Board of Reviewers

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

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

WSCG 2018
Posters Proceedings
CSRN 2803

Contents

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

Human Action Recognition from RGBD Videos based on Retina Model and Local Binary Pattern Features

Rawya Al-Akam
Active Vision Group, AGAS
Institute for Computational Visualistics
University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
rawya@uni-koblenz.de

Salah Al-Darraji
Department of Computer Science
University of Basrah
Basrah, Iraq
aldarraji@uobasrah.edu.iq

Dietrich Paulus
Active Vision Group, AGAS
Institute for Computational Visualistics,
University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
paulus@uni-koblenz.de

ABSTRACT

Human action recognition from the videos is one of the most attractive topics in computer vision during the last decades due to wide applications development. This research has mainly focused on learning and recognizing actions from RGB and Depth videos (RGBD). RGBD is a powerful source of data providing the aligned depth information which has great ability to improve the performance of different problems in image understanding and video processing. In this work, a novel system for human action recognition is proposed to extract distinctive spatio and temporal feature vectors for presenting the spatio-temporal evolutions from a set of training and testing video sequences of different actions. The feature vectors are computed in two steps: The **First** step is the motion detection from all video frames by using spatio-temporal retina model. This model gives a good structuring of video data by removing the noise and illumination variation and is used to detect potentially salient areas, these areas represent the motion information of the moving object in each frame of video sequences. In the **Second** step, because of human motion can be seen as a type of texture pattern, the local binary pattern descriptor (LBP) is used to extract features from the spatio-temporal salient areas and formulated them as a histogram to make the bag of feature vectors. To evaluate the performance of the proposed method, the k-means clustering, and Random Forest classification is applied on the bag of feature vectors. This approach is demonstrated that our system achieves superior performance in comparison with the state-of-the-art and all experimental results are depending on two public RGBD datasets.

Keywords

Action Recognition, RGBD videos, Local Binary Pattern, Retina Model, Random Forest.

1 INTRODUCTION

Human action recognition from videos is a very important research topic in image processing, computer vision, and pattern recognition. The human action recognition systems analyze the image sequences or videos by using different methods to predict the type of action and characterize the behavior of persons. This field has represented a challenge because it works with performance issues based on low illumination, perspective effects, and occlusions. It also can handle variations of people, scenes, and motion characteristics [AA13]. An efficient recognition of actions is used in many applications such as video surveillance, robotics human-computer interaction, gesture recognition, behavior analysis, and a variety of systems that involve interactions between persons and computers. All these application domains have own demands, but generally, algorithms have the ability for detecting and recognizing several actions in real time. The designed algorithm should be able to handle different forms of environment and all variations in performing actions because of the different appearance and movement of people [KZP11]. In this paper, we adopt the ideas of spatio-temporal analysis and global features extraction. Global features have been used to characterize textures informa-

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.1

tion from body motions on RGBD videos. Our approach represents the human action recognition in four steps: The **First** is high frequency spatio and temporal noise removal and the detection of motion by using a spatio-temporal output from the retina channels. The **Second** uses local binary pattern (LBP) descriptor on different retinal channels such as retina Parvo (parvocellular), Magno (magnocellula) and the combination of both channels Parvo-Magno from both RGB and depth images. The **Third** has computed the histogram from LBP of each frame in the videos and combined all histogram values into a bag of feature vectors, the bag of features algorithm which encodes all the descriptors extracted from each video into a single code. And Finally, in the **Fourth** step, we use k-means clustering and Random Forest (RF) for classifying the different action from videos. The general steps of our system is illustrated in Figure 1, which that represents the structure for our action recognition method.

The rest of the paper is represented as follow, section 2 describes the related work done in this area. Section 3 explain in detail the overview of the proposed method. Section 4 represents the experimentation and results, and finally in Section 5 provides the conclusion.

2 LITERATURE REVIEW

In this literature, the human action recognition is demonstrated from different video actions by using different methods of computer vision and machine learning techniques.

There are several methods used to detect the moving object and extract the important features information from videos, one of the most efficient techniques which is used to detect the motion is Optical Flow, that is used to segment a moving object from video frames background and track it. The moving areas through the video frames can be used to detect the area of interest (motion area) [A07, LD08].

Another research study [BD01] motion energy images (MEI) and motion history images (MHI) are used for the first time as temporal templates to represent human actions and the recognition was done by using seven Hu-moments method. The motion and salient event detection also can be detected by using the Retina model. In this model, a double spatio-temporal filtering used and a good structuring of video data occur as a noise and illumination variation removal and static and dynamic contour enhancement [CsDH10]. The human action recognition is improved by using different techniques of the global features from the moving object. The human action can be represented as a histogram of oriented gradient (HOG) of motion history image (MHI) [HHLH11], the MHI is computed with differential images from successive frames of video sequences, and the HOG features are computed from these motion area. After that, the HOG features are supplied to a support vector machine (SVM) for action classification. In the other work [KZP11] a histogram of the local binary pattern (LBP) was extracted from MHI and MEI as temporal templates to represent human action and the Hidden Markov Models (HMMs) is used to represent and recognize a temporal behavior of the action. The human action recognition can also be described by using the dynamic texture feature descriptors on spatio-temporal domains [CKZP08] and this features are used for human detection to extract LBP-TOP features in spatio-temporal domains from image data, these features are used to detect human bounding volumes and to describe human movements.

3 OVERVIEW OF THE PROPOSED METHOD

This section describes the steps of the proposed system. It details the computation of feature vector from the video sequences and the recognition of the video action. Section 3.1 introduces the pre-processing that is applied to the input RGB and depth videos. Section 3.2 gives a brief description of Bag-of-Features extraction. Section 3.3 explains the k-means clustering. Section 3.4 explains the random forest classification method which is used to compute the recognition accuracy.

3.1 Pre-processing

As the first step in this system is pre-processed to the input data as shown in Figure 2. This data is represented by RGB and depth videos which contain object appearance, shape and motion characteristics. In this work, the video sequence is converted into frames and in turn into lower resolution images of 100×100, for reducing the computational complexity of the system. The depth maps data captured by the Kinect camera are often noisy due to imperfections related to the Kinect infrared light reflections [AaP17]. To remove unwanted signals (noise) in order to preserve the required details and to eliminate the unmatched edges from the depth images, lighting variation, changes in background clutters and so on, the spatial-temporal bilateral filtering and Gaussian filtering are used for pre-processing. This process is done before feature extraction.

3.2 Bag of Features Extraction

The Bag-of-Features (BoFs) [WRLD13] is the most popular technique of feature representation in videos action for recognizing the different human actions. The global feature vectors are computed from the spatio-temporal domain by depending on motion detection and texture descriptor methods. The feature vectors are computed in two steps: Motion detection from spatio and temporal images by using retina model and feature extraction from motion area based on local binary pattern (LBP) descriptor as illustrated in the following.

ISBN 978-80-86943-42-8

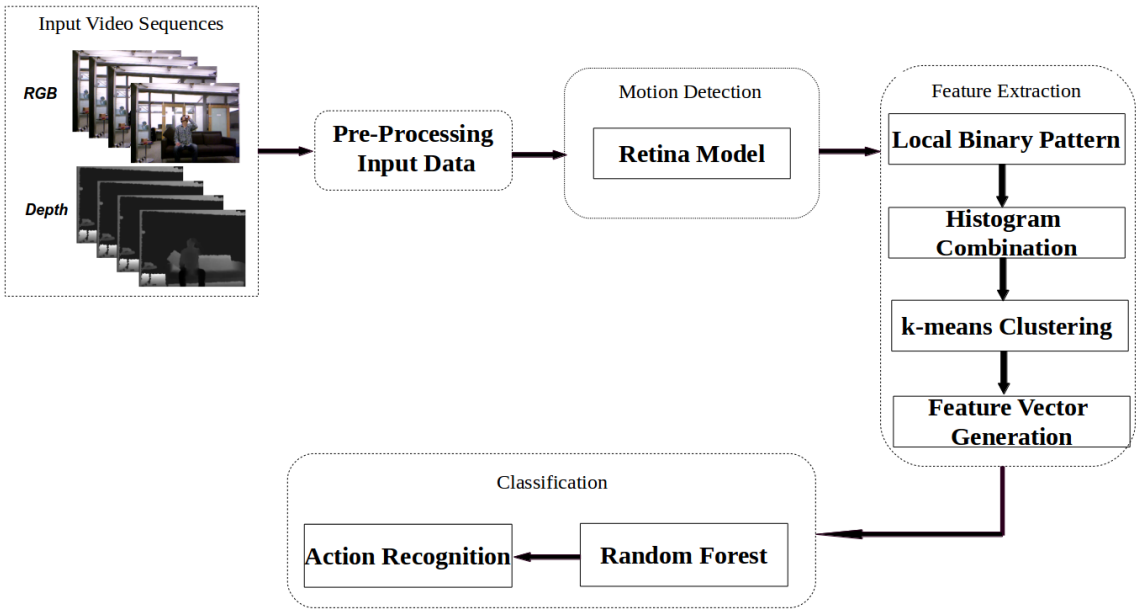


Figure 1: General structure steps of our approach for action recognition using RGB and depth video. Pre-Processing to the input video data, Motion detection, Feature extraction, and Classification.

Figure 2: Pre-processing to the input video data.

3.2.1 Motion Detection using Retina Model

The retina is a non separable spatio-temporal filter model. In order to detect the moving object in each video sequences, the human retina [1] model [CsDH10] is applied to the input data. This model is able to whiten the image spectrum and could remove the high frequency spatio and temporal noise from the images, thus providing enhanced signals for the following processing stages. The retina model decorrelation of details information of spatio and temporal by providing two output video channels, as illustrated below [SBL14]:

- The parvocellular channel (parvo), it is mainly active in the foveal retina area and provide an accurate color vision for visual details with reduced spatio-temporal high frequency noise. Furthermore, objects moving on the retina projection are blurred. The parvo retina output represented in Figure 3.
- The magnocellular channel (magno), it is fundamentally active in the retina environmental vision and send signals related to change events (motion, moving events, etc.). Also it help in improving the visual scene context and object classification from the benefits of local contrast and noise removal. The magno retina output represented in Figure 4.

Figure 3: The retina parvocellular channel (parvo), Left: RGB image and right: Depth image.

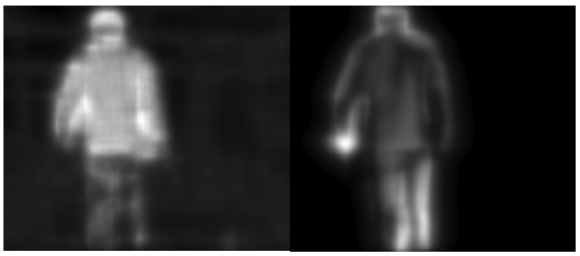


Figure 4: The retina magnocellular channel (magno), Left: RGB image and right: Depth image.

[1] https://docs.opencv.org/3.2.0/d2/d94/bioinspired_retina.html

ISBN 978-80-86943-42-8

In this work, the output of the retina model, which represents the motion area, is used to compute the global feature vectors.

3.2.2 *Features Extraction using Local Binary Pattern*

After detecting the motion area, the texture Local Binary Pattern (LBP) descriptor is used to summarize the local structures of the image [Tub17]. The illustration of the original LBP operator is shown in Figure 5, where a LBP operator is calculated by thresholding the differences among the gray value of the center pixel and the neighborhood in a 3×3 grid. Each pixel in the frame is compared with its eight neighbors. The resulting eight values are then considered as an 8-bit binary number. A binary number is obtained by concatenating all these binary codes in a clockwise direction starting from the top-left pixel and its corresponding decimal value is used for labeling. The derived binary numbers are referred to as Local Binary Patterns or LBP codes. The original LBP at the location (x_c, y_c) can be derived from this formula equation (1) as in [CKZP08], which proposed to use elliptic sampling for the x_t and y_t planes:

$$LBP_{x_c,y_c} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad s(x) = \begin{cases} 1, x \geqslant 0 \\ 0, x < 0 \end{cases} \qquad (1)$$

where g_c is the gray value of the center pixel (xc, yc) and g_p are the gray values at the P sampling points.

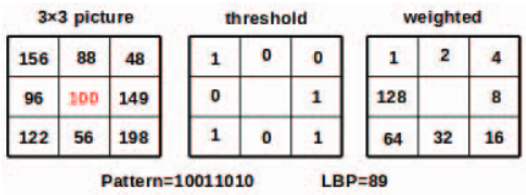


Figure 5: Illustration of the original LBP [VHS15].

The histogram of the encoded motion area is obtained by applying the LBP operator and then used as a texture descriptor for that area in the images. These LBP histograms are combined to represent the spatio-temporal feature vectors from all images in a video.

3.3 k-means Clustering

Clustering is the technique aims to divide the data into groups and each group is constructed by similar data. In simple words, the aim is to separate groups with similar type and assign them into clusters. k-means clustering is a type of unsupervised learning [Yao13], that used with unlabeled data (i.e., data without defined categories or groups) because it has a high efficiency on the data partition, especially in the large dataset. The goal of this clustering method is to find groups in the given data, with the number of groups represented by the variable k. This method is working iteratively to assign each data point to one of k groups based on the features that are provided. Data points are clustered based on feature similarity. The output results of the K-means clustering algorithm are: (1) The centroids of the k clusters, which can be used to label new data, and (2) Labels for the training data (each data point is assigned to a single cluster). In this work, after extracting all LBP features from all RGBD videos. The k-means algorithm is used to generate the dictionary from LPB feature vectors (which is called Bag of features (BOF)) and it was applied on all BoF of training videos sequences, the k represented the dictionary size. The centroids of each cluster are combined to make a dictionary. In this method, we got the best result with a value of k = 400 as a dictionary size. After that, each feature description of the video frame is compared with each centroid of the cluster in the dictionary using Euclidean distance measure (e). Then, the difference (e) was checked, if it is small or features values is close to a certain cluster, the count of that index is increased. Similarly, the other feature description of video frames are also compared and the counts of the respective indices are increased of which the feature description values are closest to which cluster[AaP17, AaP18]. These steps are computed from all the feature vectors of training and testing dataset.

3.4 Action Recognition with Random Forest

To recognize the human actions, the classifiers is needed. The Random Forest (RF) classifiers are used in this work, because of the RF can handle thousands of input variables and large dataset. Moreover, the Random Forest a good performance and outperforms many other machine learning classification algorithms for action recognition [Nab17]. Random Forest was introduced by Breiman [Bre01] as a set of decision trees. For each decision tree in this forest behave like a weak classifier and combined together to compose a strong classifier. During training stages, nodes in the trees are split by randomized selection of features. This selection decreases the error rate in the forest by decreasing the correlation among trees in the forest. Finally, each random tree in the forest grows and predicts the input test data class label. The importance of variables is estimated at the end of training stage [AA13].

ISBN 978-80-86943-42-8

4 TEST RESULTS

For our action recognition experiments, we chose to use the MSR Daily Activity 3D Dataset [2] and Online RGBD Action dataset (ORGBD) [3].

The MSR Daily Activity 3D Dataset is a daily activity dataset was recorded by Microsoft and the Northwestern University in 2012, this dataset is captured by a Kinect device and focused on daily activities in the living room, there is a sofa in the scene and the camera was fixed in front of it [Wan12]. This dataset contains 16 actions and 10 subjects; each subject performs each activity in two different poses: *drinking, eating, read a book, call cell phone, writing on a paper, using laptop, using vacuum cleaner, cheer up, sitting, still, tossing paper, playing game, laying down on sofa, walking, playing guitar, stand up, and sit down*. The total number of the activity videos are 320 samples. Some example activities are shown in Figure 6 [AaP18].

The Online RGBD Action dataset (ORGBD) [YLY15] targets for human action recognition (human-object interaction) based on RGBD video data, they are recorded by the Kinect device. Each action was performed by 16 subjects for two times. This dataset contains seven types of actions which captured in the living room: *drinking, eating, using a laptop, picking up a phone, reading phone (sending SMS), reading a book, and using a remote.* as shown in Figure 7 [AaP18]. We compare our approach with the state-of-the-art methods on the same environment test setting, where half of the subjects are used as training data and the rest of the subjects are used as test data.

Figure 6: Sample frames of MSR-Daily Activity 3D Dataset.

The proposed method for computing a texture features by using LBP texture descriptor which applied on retina

[2] http://www.uow.edu.au/w̃anqing/#MSRAction3DDatasets
[3] https://sites.google.com/site/skicyyu/orgbd

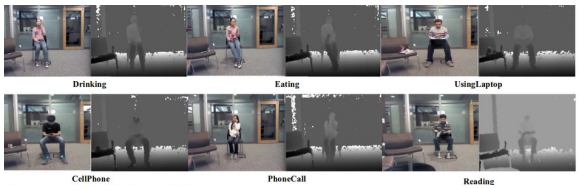


Figure 7: Sample frames of Online RGBD Action Dataset.

motion detection model from spatio-temporal domains. In this work, the experimental results are done on different retina channels as shown in Table 1, and we conclude that the best recognition results are from the LBP features on Magno retina filter (motion detection) channel in comparison with other LBP on Parvo and Combination of Parvo-Magno retinal channels. The reason for that, if the Magno channel gives a strong energy in the detected area, then the Parvo channel is certainly blurred there since there is a transient event. Table 2 and Table 3 are showing the comparison of accuracy results of our system test and the other state-of-the-art which the used different methods of the MSR-Daily Activity 3D datasets and ORGBD Dataset respectively.

In our experiments, to compute the feature vectors values three important steps is done: motion detection by using spatio-temporal retina model and the texture feature descriptor LBP is applied on retina output from both RGB and depth channels and finally the histogram from LBP are computed and combined all histogram values to form the bag of feature (BOF). The feature vector size is computed as $2^8 * 2 = 512$ from both RGB and depth channels. To compute the recognition accuracy from our system, k-means clustering and Random Forest (RF) classifier are computed from all feature vector values of different actions and gave the good accuracy rates on both types of datasets.

Retina Channels	MSR3D	ORGBD
Parvo+LBP	83.37%	93.13%
Magno+LBP	**90.21%**	**96.86%**
Parvo+Magno+LBP	81.11%	92.86%

Table 1: Our comparison of recognition accuracy using different retina channels on MSR-Daily Activity 3D (MSR3D) and Online RGBD (ORGBD) Datasets.

Methods	Accuracy
CHAR [ZZSS16]	54.7%
Discriminative Orderlet [YLY15]	60.1%
Feature covariance [PTDZ17]	65.00%
Moving Pose [ZLS13]	73.80%
Parvo+LBP	**83.37%**
Magno+LBP	**90.21%**
Parvo+Magno+LBP	**81.11%**

Table 2: Comparison of recognition accuracy with other methods on MSR-DailyActivity 3D Dataset.

ISBN 978-80-86943-42-8

Methods	Accuracy
HOSM [DLCZ16]	49.5%
Orderlet+SVM [YLY15]	68.7%
Orderlet+ boosting [YLY15]	71.4%
Human-Object Interaction[MDDB15]	75.8%
Parvo+LBP	**93.13%**
Magno+LBP	**96.86%**
Parvo+Magno+LBP	**92.86%**

Table 3: Comparison of recognition accuracy with other methods on ORGBD Dataset.

5 CONCLUSION AND FUTURE WORKS

We have presented a new method for human action recognition based on Retina model and local binary pattern (LBP) descriptor. Its main idea is to capture spatio-temporal relation of moving object depending on the spatio-temporal filtering retina model and applied the LBP texture feature extractor on the moving object from each image in video actions to compute a bag of important feature information. These feature values are tested using random forest classification machine learning methods. Our system is tested on two different public RGBD dataset and its achieved superior performance in comparison with the state-of-the-art approaches. These datasets are MSR Daily Activity 3D and Online RGBD (ORGBD) and the recognition accuracy on this dataset reached to 90.21% and 96.86% respectively. For the future work, we will apply a convolution neural network on our system.

ACKNOWLEDGEMENTS

This work was partially supported by Ministry of Higher Education and Scientific Research (MHESR), Iraq, and University of Koblenz-Landau, Germany.

6 REFERENCES

[A07] Deborah A. An Adaptive Optical Flow Technique for Person Tracking Systems. *Pattern Recognition Letters*, 25:43–53, 2007.

[AA13] Ilktan Ar and Yusuf Sinan Akgul. Action recognition using random forest prediction with combined pose-based and motion-based features. *ELECO 2013 - 8th International Conference on Electrical and Electronics Engineering*, pages 315–319, 2013.

[AaP17] Rawya Al-akam and Dietrich Paulus. RGBD Human Action Recognition using Multi-Features Combination and K-Nearest Neighbors Classification. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 8(10):383–389, 2017.

[AaP18] Rawya Al-akam and Dietrich Paulus. Local and Global feature Descriptors Combination from RGB-Depth Videos for Human Action Recognition. In *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, number Icpram, pages 265–272, 2018.

[BD01] Aaron F Bobick and James W Davis. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(3):257–267, 2001.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[CKZP08] P.A. Crook, V. Kellokumpu, G. Zhao, and M. Pietikainen. Human Activity Recognition Using a Dynamic Texture Based Method. *Procedings of the British Machine Vision Conference 2008*, pages 88.1–88.10, 2008.

[CsDH10] ABenoit A Caplier sinpgfr, B Durette, and J Herault. Using Human Visual System Modeling for Bio-Inspired Low Level Image Processing. *Computer Vision and Image Understanding.*, 114, no. 7:758 – 773, 2010.

[DLCZ16] Wenwen Ding, Kai Liu, Fei Cheng, and Jin Zhang. Learning hierarchical spatio-temporal pattern for human activity prediction. *Journal of Visual Communication and Image Representation*, 35:103–111, 2016.

[HHLH11] Chin Pan Huang, Chaur Heh Hsieh, Kuan Ting Lai, and Wei Yang Huang. Human action recognition using histogram of oriented gradient of motion history image. *Proceedings - 2011 International Conference on Instrumentation, Measurement, Computer, Communication and Control, IMCCC 2011*, pages 353–356, 2011.

[KZP11] Vili Kellokumpu, Guoying Zhao, and Matti Pietikinen. Recognition of human actions using texture descriptors. *Machine Vision and Applications*, 22(5):767–780, 2011.

[LD08] Z. Lin and Larry S. Davis. A Pose-Invariant Descriptor for Human Detection and Segmentation. *Computer Vision–ECCV 2008*, 5305:423–436, 2008.

[MDDB15] Meng Meng, Hassen Drira, Mohamed Daoudi, and Jacques Boonaert. Human-object interaction recognition by learning the distances between the object and the skeleton joints. *2015 11th IEEE Int. Conf. Work. Autom. Face Gesture Recog-*

 ISBN 978-80-86943-42-8

nit., pages 1–6, 2015.

[Nab17] Mohsen Nabian. A Comparative Study on Machine Learning Classification Models for Activity Recognition. *Journal of Information Technology & Software Engineering*, 07(04):4–8, 2017.

[PTDZ17] Alexandre Perez, Hedi Tabia, David Declercq, and Alain Zanotti. Feature covariance for human action recognition. *2016 6th International Conference on Image Processing Theory, Tools and Applications, IPTA 2016*, 2017.

[SBL14] Sabin Strat, Alexandre Benoit, and Patrick Lambert. Retina enhanced bag of words descriptors for video classification. In *European Signal Processing Conference*, 09 2014.

[Tub17] Tuba Milan Simian Dana Tuba, Eva. Support Vector Machine Optimized by Firefly Algorithm for Emphysema Classification in Lung Tissue CT Images. In *25. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG*, 2017.

[VHS15] Domonkos Varga, László Havasi, and Tamás Szirányi. Pedestrian detection in surveillance videos based on CS-LBP feature. *2015 International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2015*, (June):413–417, 2015.

[Wan12] Junsong Wang Jiang Liu Zicheng Wu Ying Yuan Junsong Wang, Author. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.

[WRLD13] Jun Wan, Q Ruan, W Li, and S Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14:2549–2582, 2013.

[Yao13] Yang Yu Yongqing Xu Hong Lv Weiming Li Zhao Chen Xiaoyun Yao, Yukai Liu. K-SVM: An effective SVM algorithm based on K-means clustering. *Journal of Computers (Finland)*, 8(10):2632–2639, 2013.

[YLY15] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9007:50–65, 2015.

[ZLS13] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. *The IEEE International Conference on Computer Vision (ICCV)*, pages 2752–2759, 2013.

[ZZSS16] Guangming Zhu, Liang Zhang, Peiyi Shen, and Juan Song. An Online Continuous Human Action Recognition Algorithm Based on the Kinect Sensor. *Sensors (Basel, Switzerland)*, 16(2):161, 2016.

A new model driven architecture for deep learning-based multimodal lifelog retrieval

Fatma Ben Abdallah
Regim-LAB
REsearch Groups in Intelligent Machines,
University of Sfax,
National Engineering School of Sfax (ENIS)
3038, Sfax, Tunisia
ben.abdallah.fatma@ieee.org

Ghada Feki
Regim-LAB
REsearch Groups in Intelligent Machines,
University of Sfax,
National Engineering School of Sfax (ENIS)
3038, Sfax, Tunisia
ghada.feki@ieee.org

Anis Ben Ammar
Regim-LAB
REsearch Groups in Intelligent Machines,
University of Sfax,
National Engineering School of Sfax (ENIS)
3038, Sfax, Tunisia
anis.ben.ammar@ieee.org

Chokri Ben Amar
Regim-LAB
REsearch Groups in Intelligent Machines,
University of Sfax,
National Engineering School of Sfax (ENIS)
3038, Sfax, Tunisia
chokri.benamar@ieee.org

ABSTRACT

Nowadays, taking photos and recording our life are daily task for the majority of people. The recorded information helped to build several applications like the self-monitoring of activities, memory assistance and long-term assisted living. This trend, called lifelogging, interests a lot of research communities such as computer vision, machine learning, human-computer interaction, pervasive computing and multimedia. Great effort have been made in the acquisition and the storage of captured data but there are still challenges in managing, analyzing, indexing, retrieving, summarizing and visualizing these captured data. In this work, we present a new model driven architecture for deep learning-based multimodal lifelog retrieval, summarization and visualization. Our proposed approach is based on different models integrated in an architecture established on four phases. Based on Convolutional Neural Network, the first phase consists of data preprocessing for discarding noisy images. In a second step, we extract several features to enhance the data description. Then, we generate a semantic segmentation to limit the search area in order to better control the runtime and the complexity. The second phase consist in analyzing the query. The third phase which based on Relational Network aims at retrieving the data matching the query. The final phase treat the diversity-based summarization with k-means which offers, to lifelogger, a key-frame concept and context selection-based visualization.

Keywords

Lifelogging, Multimodality, Retrieval, Summarization, Visualization, Convolutional Neural Network , Relational Network.

1 INTRODUCTION

Recently, we have witnessed the emergence of user-centric approaches in multimedia retrieval [Fek16] [Fak16] [Bouh17] [Gue11] [Wal10]. In fact, personalizing the search is the main objective in several on-going research domains like egocentric vision, self-tracking, quantified-self and personal data which are more commonly known for the last decade as lifelogging. Indeed, lifelog consists of acquiring data via cameras and sensors and storing this data to form a personal archive [Gur14]. Since the dawn of time, men have always tried to leave traces of their activities and their daily lives. Prehistoric men painted frescoes in the caves. Later, men recorded their thoughts, their moods and their days in diaries. Nowadays, a lot of wearable cameras have been created to facilitate the automatic capture of images and videos of daily life. In this work, we focused on data captured with photographic camera commonly called visual lifelogs, because with this kind of camera we can acquired over long periods of time which would not be possible with videos cameras. If lifelog's primary goal is to build a personal archive that extends over years, the best way to do this is to use images instead of videos. Lifelogging is characterized by the huge amount of personal data generated by the lifelogger. This data does not contain neither annotations nor semantic descriptions. An effort has been done these last years to construct lifelog datasets which contains tens of thousands of images. Considering this huge amount of personal data created, there is a need for systems that can automatically analyse the data in order to understand, classify, summarize and also query to retrieve the information the user may need. Another

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.2

specificity concerns the images, they are captured automatically at regular intervals without knowing on what focus lifelogger's attention. Sometimes, images are noisy, error-prone, blurry, distorted or contains useless information like sky or walls. Operate a preprocessing is necessary to eliminate noisy information. To make this personal data usable, several personal lifelogging applications (PLA) have been realized [Gur14]. These PLA cover different life aspect of lifelogger, self-monitoring of activities (sport, dietary, sleeping, smoking cessation), memory assistance (help people with memory loss such as with Alzheimer's disease or dementia) and long-term assisted living (to prevent older adults from potential alarm situations).Despite the advances in the visual lifelogs capture and storage, relevant data mining from this considerable amount of multimodal data remains unresolved problem. To the best of our knowledge, three major issues are present in lifelogging. First, the multimodality is generally not addressed [Dan17]. Second, a complete system combining annotation, retrieval, summarization and visualization is not proposed yet. Third, the architecture of existing systems are not based on any model.

In this paper, we present a new model driven architecture for deep learning-based multimodal retrieval, summarization and visualization. Our proposed architecture consists of four phases. The first phase process begin with preprocessing the lifelog images using CNN. Then, an extraction feature with enhancement is operate relying on several CNN pretrained on Imagenet. After that, a semantic segmentation using Global Convolutional Network (GCN) limit the search area in order to better control the runtime and the complexity. The second phase, based on Relational Network (RN), consist in retrieve moments according to the user's query. The third phase summarize the output of retrieval based on diversity using convolutional k-means. The final phase gives the summary's visualization based on different concepts and contexts. The remainder of this paper is divided into five sections. In section 2, we present recent related works in the retrieval, summarization and visualization context and discuss on-going challenges in lifelogging. In section 3, we describe our model driven approach which is based on several conceptual model. Section 4 details the four phases of our new architecture which based on several deep learning method for multimedia lifelog retrieval and summarization. The section 5 provides some concluding remarks and suggests future works.

2 OVERVIEW OF CHALLENGES IN LIFELOGGING

For several years, proposing systems which are able to extract relevant information from lifelog data has been the interest of various researchers. This necessity is linked to the exponential growth of the data recorded by the various cameras and sensors.

Lifelogging has become a full-fledged task at international conferences and a special attention is paid to it. We are witnessed the creation of tasks and benchmarks in international conferences dedicated to lifelogging such as NTCIR [Gur17] and IMAGECLEFLifelog Task [Dan17] which gave rise to the construction of test collection for lifelog research. In the following, we will present an overview on retrieval, summarization and visualization lifelog approaches which, most of them, were proposed on this benchmarks.

2.1 Deep Learning based approaches of retrieval

The central theme of the retrieval is the study of models and systems of interaction between human users and corpus of digital documents in order to satisfy their information needs. In the context of lifelogging, information retrieval consists in finding an event, an object, a person, a place or an action in a huge personal archive. Our study focused on works based on deep learning in the context of egocentric image retrieval. We found that the majority of the studied works relied on pretrained CNN.

Authors in [Rey16] design a system based on Bag-of-Words framework able to help users to find their personal objects once they have forgotten or lost. The classification of the relevant candidates and the discarded ones is achieved by thresholding. In [Oli16b], authors proposes a text-based search engine approach on certainty score tf-idf for egocentric images retrieval based on CAFFENET[1] that takes advantages of the inverted index approach. In [Oli17], authors create an automatic method based on LSDA[2] framework for semantic indexing and retrieval of wearer activities in egocentric images based on integrating heterogenous information from images and metadata. They used the retrieval engine indexation based on certainty score developed in [Oli16b]. Authors in [Oli16a] design an interactive systems which employed a semantic content tagging mechanism based on the retrieval system open source LUCENE[3] image retrieval engine. Authors in [Xia16] based their work on feature expansion using Wordnet and a manually performed query expansion by an expert. [Saf16] used three Deep Convolutional Neural Network models (Alexnet, Googlnet and VGGnet) learned on the IMAGENET[4] corpus and Multiple SVM approach learned on the TRECVID2013[5] data using the CAFFE framework. Authors in [Lin16] proposed a textual approach based on word to vector model. They

[1] http://caffe.berkeleyvision.org/
[2] http://lsda.berkeleyvision.org/
[3] http://www.lire-project.net/
[4] http://www.image-net.org/
[5] https://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html#tv13

ISBN 978-80-86943-42-8

use a word distance between the provided CAFFE concepts and the keywords from the quey. [Zhou17] used the human-in-the-loop methods to match between query and user needs.

By analyzing these works more closely, there are some limitations which we detail in the following. In [Rey16], bag of words ignores the context of words : it does not take into account semantic meaning and ordering. In [Oli16b], the authors used EDUB dataset 2015 to evaluate the approach. The dataset contain only 4192 images. With a bigger dataset, there will may be a problem of performance with the inverted index. That's why they used NTCIR-12 lifelog dataset in [Oli17]. Concerning [Oli16a], we have doubts about system adaptation to a change in the order of demand's magnitude, particularly in maintaining functionality and performance in high demand. [Xia16] not achieve promising result, they failed to provide good retrieval effectiveness as they said in their article. Authors in [Saf16] use temporal indexing which is fuzzy and culturally dependent. They also link the terms of the topic manually to the set of IMAGENET and TRECVID and generate manually the queries from the topics. In [Lin16], authors faces difficulties to construct the relations of topic question keywords and CAFFE concept words. [Zhou17] exploit only the information's provided which consist in the description of the semantic locations and physical activities.

2.2 Clustering based summarization

The summarization process aims to produce a concise version of one or more digital documents containing only the most important information, possibly responding to a user need. Ideally, information retrieval should use the abstract to present synthetic results requiring minimal time to be appreciated by the user. Summarization deals with unsupervised classification since we do not know in advance the number of classes which represent the image returned by the system. Two methods were used in the majority of the works presented below : hierarchical clustering and k-means.

In [Mol16], they filter uninformative images by analyzing their ratio edges and describes the images using the available CNN models for objects and places with egocentric-driven augmentation. Then, they cluster into episode using k-means approach. [Bol15] based their work on frames characterization by the means of the pretrained CAFFENET convnet, which will be segmented based on unsupervised hierarchical agglomerative clustering. Then to choose the best photo to represent the event , they select the most visually similar with the rest of the photos in the same cluster based on random walk and minimum distance. [Lid15] remove non-informative images by a new CNN-based filter. Then, images are ranked by relevance to ensure semantic diversity : relevance ranking was obtained by integrating techniques for saliency detection, object recognition and face detection. Moreover, re-rank is applied by enforcing diversity among the chosen subset of pictures. Finally, they define the priorities of the different relevance terms based on Mean Sum of Maximal Similarities. [Dog17] analyze the output of concept detector provided by the organizers and selecting for each image the most probable concepts. Then, they perform image clustering based on the histogram of oriented gradients (HOG) and stopped the hierarchical clustering algorithm when 30 clusters were formed. After that, they use WU-Palmer similarity score using Wordnet to calculate similarity be-tween each image from the cluster and the topic description. Finally, they sorted the clusters in descending order based on the mean value of the similarity scores of the images that it contained. [Mol17] choose to begin with preprocessing techniques to filter out uninformative images. Then, they rank the remaining images according to how well they match the given query. After that, they cluster the top ranked images into a series of events. Finally, they select images in interactive manner according to distance to the cluster for k-means clustering or relevance score for hierarchical trees. [Zhou17] use hierarchical clustering and select the top image that close to center for summarization. By looking at these works more attentively, we notice that [Mol16] and [Bol15] did not take into account diversity in the ranking. For [Mol16], the evaluated dataset is not big and representative enough and the pretrained models that they use do not properly classify egocentric images content. [Bol15] did not use semantic information like object, people or actions and rely solely on low-level features. [Lid15] did not take into account spatial and temporal information. [Dog17] provide results not satisfactory due to the lack of correlation between the concept output by the CAFFE concept detector and the meaning of textual descriptions of the topics. Besides, temporal information has not been used. They, relied solely on the information provided by the organizers and no additional annotations or external data have been used. For [Mol17], different tasks require different summarization method which may not be completely consistent when changing the lifelog input.

2.3 Multimodal Visualization

Managing, searching and browsing a large amount of lifelog images through an interface has been the subject of several research. [Lee08] was the first to create a web interface for browse, search, annotate or save for future reference Sensecam photos. [Oli16b] proposed a web based prototype too. [Oli16a] realize a heat map which sort the images by the time and highlight those that express the context of the moment. [Hwa13] propose a mobile life browser, called MylifeBrowser, which visualizes and searches the lifelog data from mo-

 ISBN 978-80-86943-42-8

bile device. [Lar13] propose quantified self (QS) Spiral an interactive visualization technique that aims to capture the quantified self-data and let the user explore those recurring patterns. [Hop13] present different visualization techniques like Comic-book style Visual diary inspired by the squarified treemap pattern, timeline, master detailed having the appearance of a thumbnail gallery, a social interaction radar graph and a activity yearly calendar. [Dua17] investigate in virtual reality by realizing a virtual reality lifelog prototype.
The works presented in the previous subsection summarization did not deal with visualization, they only display the result at screen. They do not offer a way to visualize a relevant images selection to a specific search. Indeed, generally the images captured using wearable camera can be consulted via the application on computer or on mobile phone. In this kind of application, images are sorted according to a timeline per day or where every picture was taken based on GPS sensor.
At the end of this overview, we can say that no system propose to combine annotation, retrieval, summarization and visualization given the colossal difficulty that this induces, the systems focus only on one or two of this phases but not on the all. Also, we have noticed that many works proposed systems or frameworks which are not based on any model. Furthermore, deep learning is used only for extraction feature. The majority of the proposed works relied solely on the information provided by the organizers and no additional annotations or external data have been used. In all the above mentioned approaches, performance can be improved when the CNN is retrained on images that are more related to the retrieval dataset according to [Bab14]. The majority of them trained the images on IMAGENET. They have resorted to manual annotation to fill this information gap. We focus on deep learning-based approaches since it achieved promising results compared to classical ones as they mentioned in [Kri12] and [Gar17].

3 NEW ARCHITECTURE FOR DEEP LEARNING-BASED MULTIMODAL LIFELOG RETRIEVAL, SUMMARIZATION AND VISUALIZATION

In the following, we detail the novel architecture for deep learning-based multimodal lifelog retrieval, summarization and visualization. We first describe the general architecture, then we detail every step.

3.1 Model driven

Existing works have focused on one problem at time : they treated either retrieval, summarization or visualization. But none of them, except [Zhou17] who combined retrieval and summarization, has tried to create a whole chain of processing starting from the data's preprocessing until the visualization through the retrieval and the summarization. To be able to integrate these different phases, we felt that it was essential to be based on a model. The model is an essential condition for realization of a solid architecture as well as a good understanding of the system to be developed. A model can be considered as an abstraction of a system in the form of a set of facts. We choose to use model driven approach to realize our system. To do this, we will use UML language which is considered as a platform-independent modeling language and used in model-driven architecture introduced by the Object Management Group. UML model systems according to different points of view, static and dynamic. The static structure allows to model a system using objects, attributes, operations, and relationships. We described this static structure through the use case and the class diagram. We chose to model the dynamic behavior and the interactions between objects through the activity diagram. To elaborate this diagrams, we rely on the 5R's described in [Sel10] : recollecting, reminiscing, retrieving information, reflecting and remembering intentions. More details are shown in Fig 1.

3.1.1 Metamodel for Lifelogger

The lifelogger is the main actor of our system. He's who will create the database by capturing images using a device such as a sensor, camera or smartphone. Each image will have as unique identifier the date on which it was taken. When the lifelogger wants to search for a moment, he may be able to search for an object, a person, an event, a place, an emotion or an activity daily living (ADL) context and concept-based composed of individual actions. The result should be summarized before visualization. The fig.2 shows the map for modelling lifelogger needs.

3.1.2 Activity diagram

The activity diagram is a dynamic UML diagram describing the sequential activities and parallel systems. They allow to represent graphically the behavior of a method. The fig.3.a described the sequencing of a captured image and a search for a moment through the system. Following the preprocessing, the image can be deleted if it is uninformative. If the image is blurred, the system try to unblur it. If the image contain homogenous color this will mean that the image is uninformative and it will be deleted. Then, the system extract image feature and try to detect concepts. If the system find a new concept, it then make automatic annotation to the image. In both case, when finding or no a new concept, the next step is to realize a semantic segmentation. After that, when the lifelogger submit a query to find a specific moment, the system active the retrieval phase. Afterward, the system rank the result

 ISBN 978-80-86943-42-8

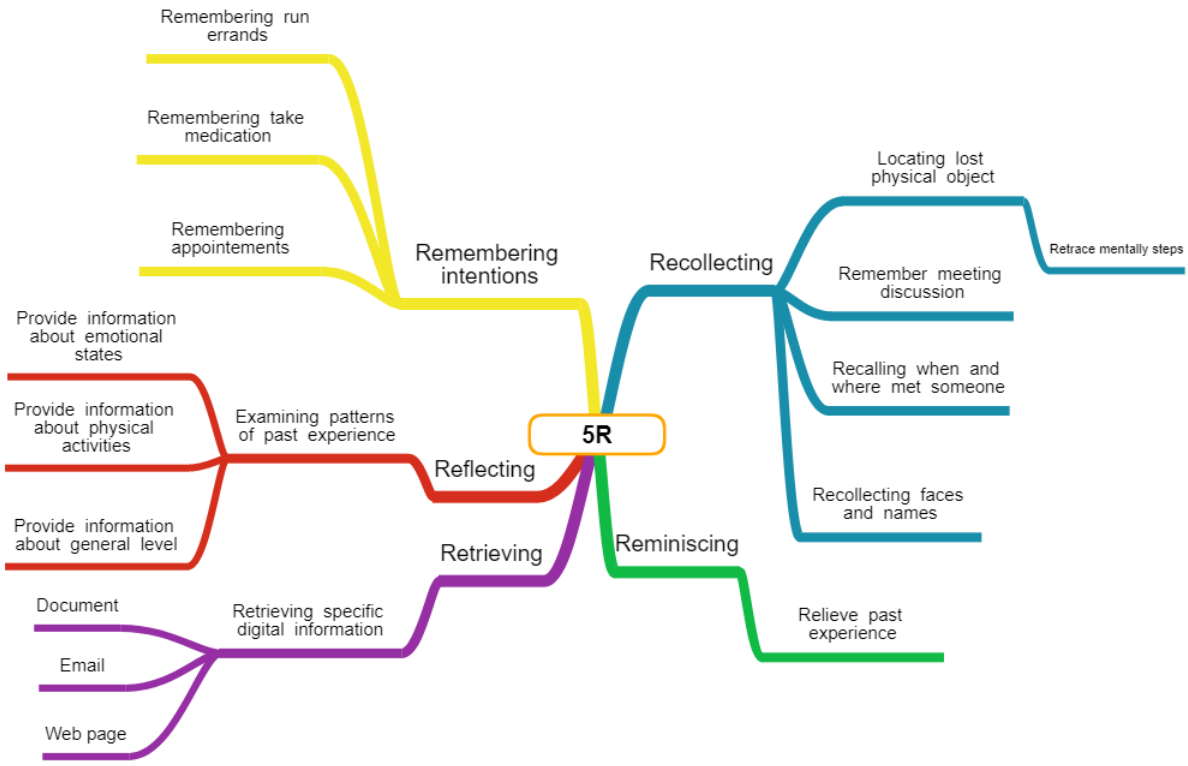


Figure 1: 5Rs mind map

of the retrieval and summarize it. The outcome is then visualize by the lifelogger.

3.2 Proposed architecture

Deep learning (DL) uses supervised learning based on digital artificial neural networks to allows a program, for example, to recognize the content of an image [Fak17] or to understand spoken language like Siri, Cortana and Google Now. With traditional methods, the machine simply compares the pixels. DL allows learning on more abstract characteristics than pixel values, which it will auto-construct. The knowledge on the classification of images contained in such network can be exploited in two ways: as an automatic extractor features, materialized by the CNN code and as a Fine Tuning to deal with the new classification problem. Given the power of DL in recognition and image processing [Boug14], we chose to use this method in our architecture.

The input of our architecture is a lifelog dataset which contains images, images concepts, extra data and queries. These multimodal and heterogeneous data was recorded from several acquisition modalities and came in different formats. The first one contain three steps: preprocessing, image feature extraction and semantic segmentation. This phase aims to delete uninformative image, to unblur blurred image by using pre-trained CNN as a feature extractor and to group images in homogeneous segment. We notice that performance can be improved when the CNN is retrained on images that are more related to the retrieval dataset. We will then also work on training the CNN on annotated lifelog dataset to improve performance and relevance. In the second phase, we use LSTM encoder to realize query processing. Since we will use the Relational Network to solve the retrieval problem in the third phase, it is better to perform this encoding. After ranking the result returned by the retrieval step using convolutional k-means, our architecture summarize the result based on diversity using keyframe selection. Finally, the result is shown to the lifelogger in a personalized way based on concepts and contexts. In the following, we will detailed every step includes in the architecture shown by the fig.3.b.

3.2.1 Automatic annotation enhancement

The images captured using wearable camera have a particularity. Indeed, these images are captured automatically at regular intervals (every 30 seconds) which causes repetitive shooting. Moreover, these images are sometimes taken in bad conditions like bad lighting or bad framing which give blurry images. To judge the quality of an image, we choose a CNN for no-reference image quality assessment [Bos16] applied to blurriness

ISBN 978-80-86943-42-8

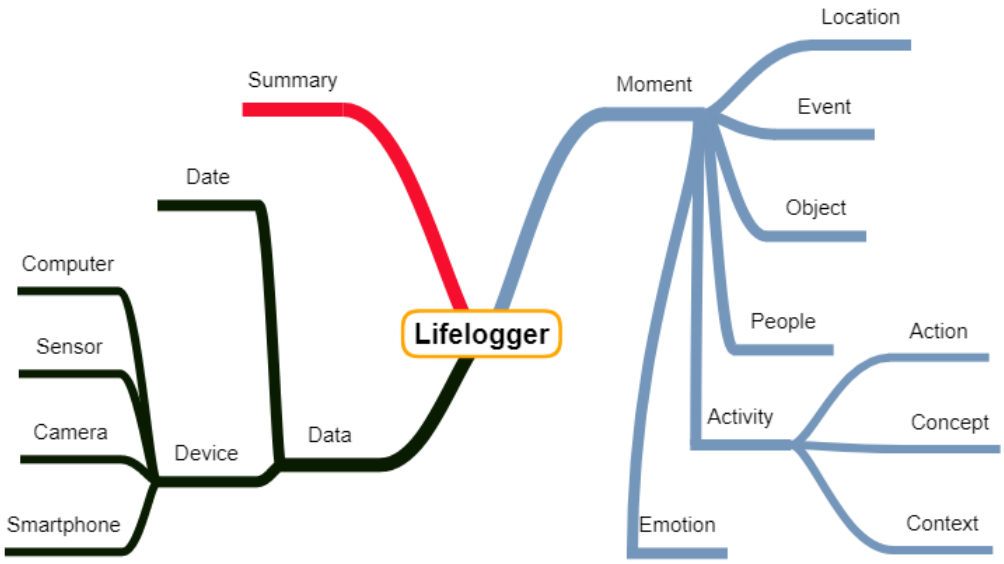
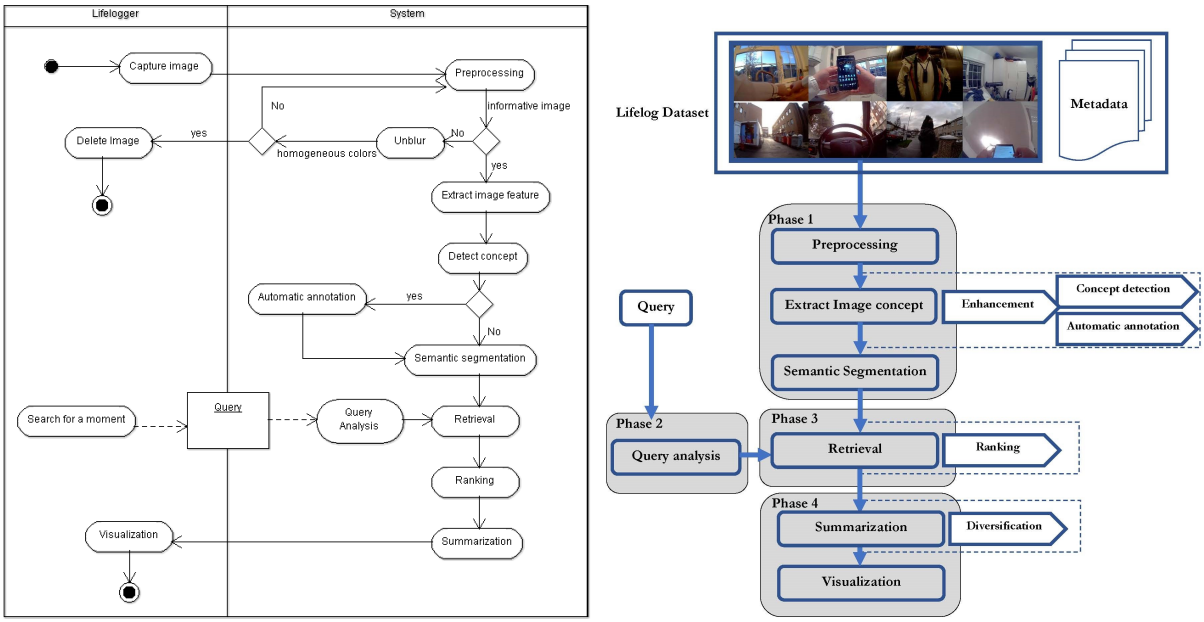


Figure 2: Lifelogger needs map



(a) Activity diagram of lifelog retrieval, summarization and visualization system

(b) Proposed architecture

Figure 3: General architecture

and colour diversity. If the image is blurred, we try to unblur it otherwise if the image contains homogeneous colour this means that the image is uninformative and should be deleted because it may be contain a sky, a wall or a floor.

The image is no longer considered as a matrix of colours' pixels, but as a carrier of a semantics encompassing several concepts. Convolutional neural networks have made considerable progress in the analysis of images especially on large dataset. To extract image feature, we rely on several CNN trained on Imagenet. In this step, we proceed to enhancement through a combination of concept detection by using the LSDA object detector which transfer classifiers for categories into detectors and automatic annotation if the discovered concept in the image do not appear in the list of concepts already related to the image in the lifelog dataset.

The objective of the semantic segmentation is the association of each object of the image, a label among a set of predefined classes (human activities, food, computer activity, heart rate ...). The final goal is to predict a mask of segmentation that indicates the category of each object. Pixels are classified based on

ISBN 978-80-86943-42-8

characteristics extracted in the image feature extraction step. By forming segments, we reduce the number of images to be processed in the retrieval phase. For that purpose, we will use a global convolutional network described in [Pen17] based on scalable ontology driven framework for hierarchical concept detection.

3.2.2 Query analysis

In the Lifelog Semantic Access Task (LSAT) of NTCIR-12 and in the lifelog retrieval task (LRT) of IMAGECLEF2017, the query is a set of topics representing lifelogger's information needs. Therefore, it is necessary to filter this query in order to extract the key concepts based on the following axes: object, location, event, ADL, people or emotion. Considering that we will use the Relational Network [San17] in the retrieval phase, the query should go through LSTM which it is capable of learning long-term dependencies.

3.2.3 Image retrieval

Information retrieval models define a representation of documents and the queries as well as a correspondence function which makes it possible to calculate similarities between documents and queries and to rank the results. Whatever the research model and the calculations are purely numerical and rely essentially on the frequency of words and analysis of their distribution, the search for semantic information seeks to go beyond this approach by injecting knowledge [Fek15]. For that purpose, we choose to adapt Relational Network (RN) by using Neural Tensor Network instead of Multi-layer Perceptron. To use RN, we need to build a reasoning lifelog dataset, which contain images and questions that test logical reasoning to enable detailed analysis of visual reasoning. We will then rank the result using Support Vector Machine.

3.2.4 Summarization and Visualization

The information's relevance returned to the lifelogger is an important aspect of the information retrieval but we should also take into account the diversity. Images sorted by relevance may be similar and redundant. It is not interesting to rank all similar image of the same one, even if it may be the most relevant. Instead, the top results should be different and complementary. Once the images are ranked, we use a k-means clustering based on average distance to apply diversification [Fek17] [Fek14] [Ksi13] on the result that will be visualized. The top n images from each cluster will be selected where n is a limit fixed by the user.

According to the user-study done by [Cho16], visualization should be insightful, intuitive, interactive, impressive and immersive. Also, the studied lifeloggers prefer a visualization based on the most frequently visited locations in visually appealing and informative interface that another based on temporal clustering, tempo-spatial clustering or tempo-spatial-visual clustering. We offer flexible visualization's interface in order to respond to various user needs. We also investigate in describing as the frequency and spending time for activities of daily living concepts (exp. : commuting , travelling, preparing meal, ...) and total time for contexts (exp. : in an office environment, in a home, in an open space, ...).

4 EXPERIMENTS

To evaluate the performance of our method, we employed ImageCLEFlifelog2017 which is based on the data available for the NTCIR12Lifelog task. ImageCLEFlifelog2017 was gathered by 3 lifeloggers during one month giving 79 days of data. It contains 88 124 images acquired using OMG Autographer wearable camera, XML description of semantic locations and the physical activities of each lifelogger at one minute. The output of the CAFFE CNN-based visual concept detector was included in the test collection.

We realized a preliminary experimentation for the automatic annotation enhancement using Matlab neural network toolbox. To extract image concept, we rely on four CNN : VGG-19, Resnet-50, Resnet-101 and InceptionV3 trained on Imagenet. The concept detection is based on CNNs choice which was guided by an analysis of deep neural network models for practical applications [Can16]. The analysis compare AlexNet, BN AlexNet, BN NIN, ENet, GoogleNet, VGG-16, VGG-19, Resnet-18, Resnet-34, Resnet-50, Resnet-101, Resnet-152, InceptionV3 and Inception V4 in term of accuracy, parameters, memory footprint, power consumption, operations count and inference time. According to the results of the analysis, we focus on the three top CNN that have provided the best result in accuracy.

An example of concept detection is shown in fig. 4. We split the image because the input of the VGG19, Resnet50 and Resnet101 CNN should be an image with frame size of 224x224. For InceptionV3, the frame size should be 299x299.

The table 1 details the relevance of each detected concept for the corresponding frame. We use several CNNs with different numbers of layers. The CNN layers consist of convolutional layers, pooling layers, fully connected layers and normalization layers. We notice that InceptionV3, which contains 316 layers, generates 5 relevant concepts. In fact, the concept "Coffee Mug" is detected thanks to the wider frame size compared to the frame size of the Resnet-50, Resnet-101 and VGG-19. This object is not detected since it is located on the border by the other CNN. For the architecture which have the same frame size, we notice that Resnet-101 which contains 347 layers, generates 2 out of 20 relevant concepts.

 ISBN 978-80-86943-42-8

(a) With VGG19, Resnet50 and Resnet101 (b) With Inception V3

Figure 4: Example of concept detection

Frame	Resnet50	R	I	Resnet101	R	I	VGG19	R	I	InceptionV3	R	I
1	Groom		√	Groom		√	Mosquito net		√	Coffee Mug	√	
2	Wall clock		√	Book jacket		√	Ipod		√	Loud Speaker		√
3	Notebook		√	Ipod		√	Notebook		√	Chime		√
4	Refrigerator		√	Wash basin		√	Chime		√	Photocopier		√
5	Photocopier		√	Refrigerator		√	Payphone		√	Medecine chest		√
6	Beer bottle	√		Laptop		√	Rubber eraser		√	Computer Keybord		√
7	Cash machine		√	Medecine chest	√		Ipod		√	Banjo		√
8	Shoji		√	Cash machine		√	Notebook		√	Casette player		√
9	Wash basin		√	Dish washer		√	Window shade		√	Toaster	√	
10	Photocopier		√	Stove		√	Space bar		√	Coffepot	√	
11	Doormat		√	Photocopier		√	Laptop		√	Frying pan	√	
12	Tub		√	Radiator		√	Prison		√	Watter bottle	√	
13	Carton		√	Tub		√	Banjo		√			
14	Wash Basin		√	Tub		√	Prison		√			
15	Cassette		√	Wash basin		√	Tape player		√			
16	Bannister		√	CD player		√	Bannister		√			
17	Coffepot	√		Planetarium		√	Coffepot	√				
18	Coffepot	√		Coffepot	√		Vaccum		√			
19	Nipple		√	Thimble		√	Oil filter	√				
20	Water bottle	√		Nipple		√	Pill bottle	√				

Table 1: Top20 concepts detected with different CNNs(**R**:Relevant,**I**:Irrelevant)

Although, it contains 177 layers, Resnet-50 generates 4 out of 20 relevant concepts. The reason for this difference is due to the single-crop error rates : models perform better when using more than one crop at test-time. Similarly, VGG-19 generates 3 out of 20 relevant concepts with 47 layers.

5 CONCLUSION

In this paper, we have addressed the creation of a novel model driven architecture for deep learning-based multimodal lifelog retrieval, summarization and visualization. The architecture consist of four phases integrating several conceptual models. The first phase process begin with preprocessing the lifelog images using CNN.

ISBN 978-80-86943-42-8

Then, an extraction feature with enhancement is operate relying on several CNN trained on Imagenet. After that, a semantic segmentation, using GCN, limit the search area in order to better control the runtime and the complexity. The second phase, based on RN, consist in retrieve moments according to the user's query. The third phase summarize the output of retrieval based on diversity using convolutional k-means. The final phase gives the summary's visualization based on different concepts and contexts. As future works, we are making implementation of the several phases of our system and we aim to validate our architecture by participating at IMAGECLEF 2018 Lifelog Task.

6 ACKNOWLEDGMENTS

The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

7 REFERENCES

[Bab14] Babenko A., Slesarev A., Chigorin A., Lempitsky V. Neural Codes for Image Retrieval. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision - ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham.

[Bol15] Bolanos M., Mestre R., Talavera E., Giro X., Radeva P., Visual summary of egocentric photostreams by representative keyframes, ICME Workshops, pp. 1-6, 2015.

[Bos16] Bosse S., Maniry D., Wiegand T., Samek W.:A deep neural network for image quality assessment. ICIP 2016:pp 3773-3777, 2016.

[Boug14] Boughrara H., Chtourou, M., Ben Amar C., & Chen, L., Facial expression recognition based on a mlp neural network using constructive training algorithm, Multimedia Tools and Applications, pp. 709-731, 2014.

[Bouh17] Bouhlel N.; Feki G.; Ben Ammar A. & Ben Amar C., A Hypergraph-Based Reranking Model for Retrieving Diverse Social Images , Computer Analysis of Images and Patterns, Springer International Publishing, pp. 279-291, 2017.

[Can16] Canziani A., Paszke A. & Culurciello, E., An Analysis of Deep Neural Network Models for Practical Applications, CoRR, 2016 , abs/1605.07678.

[Cho16] Chowdhury S., Ferdous M. S. & Jose J. M., A user-study examining visualization of lifelogs, 2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1-6, 2016.

[Dan17] Dang-Nguyen D.-T., Piras L., Riegler M., Boato G., Zhou L. & Gurrin C., Overview of ImageCLEFlifelog 2017: Lifelog Retrieval and Summarization, CLEF2017 Working Notes, CEUR-WS, volume 1866, 2017.

[Dog17] Dogariu, M. & Ionescu, B., A Textual Filtering of HOG-Based Hierarchical Clustering of Lifelog Data, Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.

[Dua17] Duane A., Gurrin C., Investigating virtual reality as a tool for visual lifelog exploration(2017).

[Fak16] Fakhfakh R., Feki G., Ammar, A. B. & Amar C. B. Personalizing information retrieval: A new model for user preferences elicitation, IEEE International Conference on Systems, Man and Cybernetics (SMC), 2016.

[Fak17] Fakhfakh R., Ben Ammar A. & Ben Amar C., Deep Learning-Based Recommendation: Current Issues and Challenges, International Journal of Advanced Computer Science and Applications(IJACSA), 8(12), 2017.

[Fek14] Feki G., Ben Ammar A. & Ben Amar C., Adaptive Semantic Construction for Diversity-based Image Retrieval, Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1, SCITEPRESS - Science and Technology Publications, LDA, pp. 444-449, 2014.

[Fek15] Feki G., Fakhfakh R., Ammar A. B. & Amar C. B., Knowledge structures: Which one to use for the query disambiguation? 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 499-504, 2015.

[Fek16] Feki G., Feki, G., Ammar A. B. & Amar C. B. Query Disambiguation: User-centric Approach, Journal of Information Assurance Security, Vol. 11 Issue 3, pp. 144-156, 2016.

[Fek17] Feki G., Ben Ammar A. & Ben Amar C., Towards diverse visual suggestions on Flickr, Proceedings Volume 10341, Ninth International Conference on Machine Vision (ICMV 2016), 2017.

[Gar17] Garcia-Garcia A., Orts-Escolano S., Oprea S., Villena-Martinez V., Garcia-Rodriguez J. A review on deep learning techniques applied to semantic segmentation, arXiv:1704.06857, 2017.

[Gue11] Guedri B., Zaied M. & Amar C. B., Indexing and images retrieval by content, International Conference on High Performance Computing Simulation, ,pp. 369-375, 2011.

[Gur14] Gurrin C., Smeaton A. F. & Doherty A. R., LifeLogging: Personal Big Data, Foundations and Trends in Information Retrieval, pp 1-125, 2014.

 ISBN 978-80-86943-42-8

[Gur17] Gurrin C., Joho H., Hopfgartner F., Zhou L., Gupta R., Albatal R. & Dang Nguyen D. T., Overview of NTCIR-13 Lifelog-2 Task, Thirteenth NTCIR conference (NTCIR-13), Tokyo, Japan, 5-8 Dec 2017.

[Hop13] Hopfgartner F., Yang Y., Zhou L., Gurrin, C. User Interaction Templates for the Design of Lifelogging Systems, In Semantic Models for Adaptive Interactive Systems, pp. 187-204, 2013.

[Hwa13] Hwang K.S., Cho S.B., A Lifelog browser for visualization and search of mobile everyday-life, Mobile Information Systems, 10, pp. 243-258, 2013.

[Kri12] Krizhevsky A., Sutskever I., Hinton G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems, 2012.

[Ksi13] Ksibi A., Feki G., Ben Ammar A. & Ben Amar C., Effective Diversification for Ambiguous Queries in Social Image Retrieval, Computer Analysis of Images and Patterns, Springer Berlin Heidelberg, pp. 571-578, 2013.

[Lar13] Larsen J.E., Cuttone A., Jorgensen S.L., QS Spiral: Visualizing periodic quantified self-data, In proceedings of CHI 2013 Workshop on Personal Informatics in the Wild, 2013.

[Lee08] Lee H., Smeaton A.F., O'Connor N.E. et al., Constructing a SenseCam Visual Diary as a Media Process, Multimedia Systems 14: 341, 2008.

[Lid15] Lidon A., Bolanos M., Dimiccoli M., Radeva P., GaroleraM., Giro-i-Nieto X., Semantic summarization of egocentric photostream events, CoRR,http://arxiv.org/abs/1511.00438, 2015.

[Lin16] Lin H. L., Chiang T.-C., Chen L.P., Yang P.C. Image Searching by Events with Deep Learning for NTCIR-12 Lifelog. In The 12th NTCIR Conference, Tokyo, Japan, 2016.

[Mol16] del Molino A. G., Xu Q., Lim J.-H., Describing Lifelogs with Convolutional Neural Networks: A Comparative Study, LTA '16 Proceedings of the first Workshop on Lifelogging Tools and Applications, pp 39-44, 2016.

[Mol17] del Molino A. G., Mandal B., Lin J., Lim J., Subbaraju V. & Chandrasekhar V. VC-I2R@ImageCLEF2017: Ensemble of Deep Learned Features for Lifelog Video Summarization. Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.

[Oli16a] Oliveira Barra G., Ayala A. C., Bolanos M., Dimiccoli M., Giro i Nieto X. & Radeva, P., LEMoRe: A Lifelog Engine for Moments Retrieval at the NTCIR-Lifelog LSAT Task, Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016.

[Oli16b] Oliveira-Barra G., Dimiccoli M. & Radeva P., Egocentric Image Retrieval with Convolutional Neural Networks, Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, pp 71-76, 2016.

[Oli17] Oliveira-Barra G., Dimiccoli M, & Radeva P. Leveraging Activity Indexing for Egocentric Image Retrieval. In: Alexandre L., Salvador Sanchez J., Rodrigues J. (eds) Pattern Recognition and Image Analysis, IbPRIA, Lecture Notes in Computer Science, vol 10255. Springer, Cham, 2017.

[Pen17] Peng C., Zhang X., Yu G., Luo G., Sun J., Large Kernel Matters - Improve Semantic Segmentation by Global Convolutional Network, CoRR abs/1703.02719, 2017.

[Rey16] Reyes R. C. Time-sensitive Egocentric Image Retrieval for Findings Objects in Life-logs, Degree's Thesis Sciences and Telecommunication Technologies Engineering, Technical University of Catalonia, 2016.

[Saf16] Safadi B., Mulhem P., Quénot G., Chevallet J.P. LIG-MRIM at NTCIR-12 Lifelog Semantic Access Task, In The 12th NTCIR Conference, Tokyo, Japan, 2016.

[San17] Santoro A., Raposo D., Barrett D. G., Malinowski M., Pascanu R, Battaglia P., Lillicrap T., A simple neural network module for relational reasoning, arXiv preprint arXiv:1706.01427, 2017.

[Sel10] Sellen A. J. , Whittaker S., Beyond total capture: a constructive critique of lifelogging. Commun. ACM 53, pp 70-77, 2010.

[Wal10] Wali A.,Ben Aoun N., Karray H., Ben Amar C. & Alimi A. M., A New System for Event Detection from Video Surveillance Sequences Advanced Concepts for Intelligent Vision Systems, Springer Berlin Heidelberg, pp110-120, 2010.

[Xia16] Xia L., Ma Y., Fan W. VTIR at the NTCIR-12 2016 Lifelog Semantic Access Task. In The 12th NTCIR Conference, Evaluation of Information Access Technologies. Tokyo, Japan, 2016.

[Zhou17] Zhou L., Piras L., Riegler M., Boato G., Nguyen D. T. D., Gurrin C. Organizer Team at ImageCLEFlifelog 2017: Baseline Approaches for Lifelog Retrieval and Summarization, 2017.

 ISBN 978-80-86943-42-8

Human Action Recognition based on 3D Convolution Neural Networks from RGBD Videos

Rawya Al-Akam
Active Vision Group, AGAS
Institute for Computational Visualistics
University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
rawya@uni-koblenz.de

Dietrich Paulus
Active Vision Group, AGAS
Institute for Computational Visualistics,
University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
paulus@uni-koblenz.de

Darius Gharabaghi
Institute for Computational Visualistics,
University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
darius.gh@gmx.de

ABSTRACT

Human action recognition with color and depth sensors has received increasing attention in image processing and computer vision. This paper target is to develop a novel deep model for recognizing human action from the fusion of RGB-D videos based on a Convolutional Neural Network. This work is proposed a novel 3D Convolutional Neural Network architecture that implicitly captures motion information between adjacent frames, which are represented in two main steps: As a **First**, the optical flow is used to extract motion information from spatio-temporal domains of the different RGB-D video actions. This information is used to compute the features vector values from deep 3D CNN model. **Secondly**, train and evaluate a 3D CNN from three channels of the input video sequences (i.e. RGB, depth and combining information from both channels (RGB-D)) to obtain a feature representation for a 3D CNN model. For evaluating the accuracy results, a Convolutional Neural Network based on different data channels are trained and additionally the possibilities of feature extraction from 3D Convolutional Neural Network and the features are examined by support vector machine to improve and recognize human actions. From this methods, we demonstrate that the test results from RGB-D channels better than the results from each channel trained separately by baseline Convolutional Neural Network and outperform the state of the art on the same public datasets.

Keywords

Action Recognition, RGBD videos, Optical Flow, 3D Convolutional Neural Network, Support Vector Machine.

1 INTRODUCTION

The human action recognition from videos is challenging field in real-world actions and has advanced rapidly over the last few years. Due to the large intra-class variations, high dimension of video data, varying motion speed, partial occlusion and clutter background, precise action recognition is still a big challenging task. And the efficient solutions to this challenging and difficult problem can facilitate several useful applications such as visual surveillance, human-robot cooperation, and medical monitoring systems [JBCS13]. A recent development of range sensors had an incontrovertible influence on research and applications of machine and computer vision field. Sensor devices provide depth information of the scene view and objects, that helps in solving problems which are looked hard for RGB images or videos [HSXS13].

Classical action recognition tasks mainly depend on hand-crafted features which can be divided into local and global approaches. Local feature extraction methods which consist of two steps: detection and description, such as the spatio-temporal interest point detection (STIP) [Lap05], improved dense trajectories (IDT) [WS13] and histogram of optical flow (HOF) [LSR08] are widely used as a local feature for human action recognition task. Local feature extraction approaches are much more efficient and robust in real scenes applications. While the global feature extraction approaches represent the video sequence as a whole which is capturing the general appearance and motion

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.3

information from each frame in video sequences. In spite of the global approaches are very sensitive to occlusion, cluttering and shift but it is still using and commonly existing for human action recognition tasks [SLY16]. Regrettably, the hand-crafted features-based encoding methods such as fisher vector [PD07] and bag-of-words (BoW) [VVV16, CDF+04], which are represented as universal visual that does not consider much about temporal information for video-based action recognition [YCXL17].

In recent years the use of neural networks and deep learning algorithms has shown significant progress in several fundamental problems in computer vision, including action and activity recognition and also the Convolutional Neural Network (CNN) is utilized to solve different image processing tasks such as object and action recognition, and also the classification done from not only static images but also from dynamic sequence of images. And besides that, the recent development of depth camera sensors that enabled us to capture effective 3D structures of the scenes and objects [HSXS13]. This helps the vision to move from 2D towards the 3D vision, like 3D scene understanding, 3D object recognition, and 3D action recognition.

The focus of this work is improving the human action recognition from RGB and depth information by using the big public datasets. The contribution of this paper is represented in two folds: **First**) we used a 3D Convolutional Neural Network (3D CNN) model for recognizing action based on optical flow information. The CNN is used for learning high-level descriptors from low-level motion features (optical flow) by using different input video channels, such as RGB and depth information which are represented by OF-RGB-CNN and OF-Depth-CNN models; **Second**) we are examined the possibilities of feature extraction from 3D CNN and classified by multi-class support vector machine (SVM). This approach is improved that the combination of these two previous models with SVM which outperform the results of each model separately.

The rest of this paper is organized as follows. Section 2 explores the previous work related to this area. Section 3 introduces the proposed approach of the system model. Section Section 4 provides our experimental and results, and section Section 5 concludes the paper.

2 LITERATURE REVIEW

Human action recognition is one of the most interesting topics of computer vision, and it has many use cases within the academy, surveillance, robotics, games, and entertainment multimedia; because of this, the quantity and variety of works is impressive and impossible to cover in a single work. This section is focused on reviewing the works which consider being relevant to this particular problem and to our approach. There are different works which applied deep neural networks to multi-modal learning. Yosinski et. al. [YCBL14] explained a method of how the transferability of features from each layer of a neural network, and exposes their generality or specificity. Further, they evaluated and examined while a layer is general or specific to the training data, and how got good features from these layers transfer to be better than other tasks. For this purpose, several CNN is trained and the weights from different layers are transferred to other networks. Then these networks are trained again, respectively fine-tuned, using different transfer learning strategies such as freezing the weights of certain layers. Simonyan and Zisserman [SZ14] proposed a two-stream CNN architecture using multiple optical flow images computed from RGB video frames for action recognition. A temporal CNN is trained on optical flow volumes, and storing the horizontal and vertical displacement vectors from consecutive action frames, which is finally combined with a spatial CNN trained on RGB frames, to include individual scene and object features. Razavian et. al. [RASC14] improved the possibilities of feature extraction from CNN using the OverFeat network. They extracted feature vectors, respectively the network activations from a fully connected layer, and finally, they have used a support vector machine for the classification task. In addition, they selected datasets and tasks which were different from the OverFeat networks original task, e.g. classification of birds and flowers or image instance retrieval for buildings. Despite these differences, their method has achieved superior performance compared to state of the art methods and proven that pre-trained deep CNN is suitable for generic feature extraction. Athiwaratkun and Kang [Ath15] also improved the possibilities of utilizing features extracted from a pre-trained network and by depending on these features values, they evaluate the quality and performance of these vectors gained form different network layers. These feature vectors are used to train SVM [AEV17] and Random Forest classifiers which even outperform their baseline CNN.

In recent year because of the development of depth sensor, there are different works by using RGB and depth data as input to the CNN. Xinhang et. al. [SHJ17] improved the scene recognition by transmitting pre-trained RGB-CNN models and fine-tuning from RGB to the target of the RGB-D dataset. For RGB-D scene recognition, they combined RGB and depth features by projecting them in a common space and further leaning a multilayer classifier, which is jointly optimized in an end-to-end network. In the other research, A 3-CNN channels are based on rotated 3D points generated from depth maps which are introduced by Wang et. al. [PW14], and they are used the weighted hierarchical depth motion maps to store temporal motion information from different views: top, side and front views, for

 ISBN 978-80-86943-42-8

generating synthesized data, respectively rotation and temporal scaling. The three views are formed to be an input to distinct CNN. Another groups [Wan14] have demonstrated the model of 3D activity recognition from RGB-D data with reconfigurable Convolutional Neural Networks which is handled realistic challenges in 3D data, and it is enabled to perform recognition from it acts directly on the raw inputs (grayscale-depth data) to conduct recognition rather than relying on hand-crafted features. Our deep structured 3D model can be viewed as an extension of these existing approaches, in which make the network could be reconfigurable during learning and inference.

3 PROPOSED APPROACH

The proposed methodology comprises of the major states as shown in Figure 1. The 3D convolution operation is applied to extract spatio and temporal features from video data for action recognition. These 3D feature extractors operate in both the spatio-temporal domains, thus capturing motion information from video streams as illustrated at the next steps:

- The first CNN model has utilized an optical flow representation from RGB videos based on multi-frame dense optical flow (OF-RGB-CNN). This optical flow is computed to hold temporal motion information from temporal domains that fed directly to the CNN (RGB-CNN) to compute the feature vector values which are trained and evaluate directly inside CNN and also these feature vectors are testing with multi-class SVM.
- The second CNN model has used the depth data from corresponding actions of the same RGB video. In contrast to the OF-RGB-CNN approach, the depth-CNN is trained and evaluated in a similar manner.
- Finally, this 3D CNN architecture generates multiple channels of information from adjacent video frames of OF-RGB and OF-depth dimensions and performs sub-sampling and convolution separately in each channel. The final feature representation is obtained by combining information from all channels of both CNN models (OF-RGB-Depth). To explore the possibilities of feature extraction and evaluation with another classifier. The multi-class support vector machine (SVM) classifier is used for this testing evaluation. For this purpose, each CNN serves as a fixed feature extractor. The evaluation of this classification method is then done separately for each CNN model and when combined these two previous models with SVM classifier.

The general structure steps of our human action recognition system are explained at the next subsections in details.

3.1 Preprocessing

3.1.1 RGB Video Preprocessing

The original RGB datasets come in sets of "avi" videos format and have a resolution of 1920 x 1080 pixels. As a first, the video is converted to a sequence of frames and each frame is cut to quadratic size since the subjects appear mostly around the image center. Then each frame is resized to 360 x 360 pixels and for lower computation complexity each frame is converted to a grayscale image.

3.1.2 Depth Video Preprocessing

The original masked depth datasets are coming as the sets of individual frames in "png" format and have the resolution of 512 x 424 pixels. The individual image values are given in millimeters. The masked depth data is already preprocessed and extract foreground data from it. However, the masked depth data still involves challenges. Strong noise can be found in the ground area in all samples and can not easily be removed because of occlusion with feet and legs. This noise could be caused by lighting conditions or camera parameters. Each sample comes in a folder associated with sample number, action ID, camera setup and so on. To keep track of the sample order a shell script is used to sort the samples by their action ID. The frames are first cut to resolution 400 x 400 to further reduce unnecessary image space, also the quadratic shape can be beneficial for the matrix. The image values are then converted from millimeters to the range [0,255], additionally, histogram spreading is applied for better visualization. To reduce memory consumption and training time the images are finally resized to 64 x 64 pixels. The example of this process is shown in Figure 2.

3.2 Feature Extraction

3.2.1 Dense Optical Flow

Optical flow displacement data is generated to capture temporal motion information as in [SZ14, RF16] from RGB and depth data to trained a CNN. Each video sequence is divided into the pairs of consecutive grayscale frames. Then the dense optical flow is computed between each pair of consecutive frames t and $(t+1)$. For the optical flow computation, the Farneb"ack optical flow [Far03] method is applied. The output is two channels image storing the horizontal dx and vertical dy displacement of each pixel location (u,v). Maximum and Minimum values over all frames are used for image normalization. Figure 3 shows a sample optical flow volume and corresponding RGB frames. In this work, the two channel images are further resized to 64x64 pixels and then split into a vertical and a horizontal component. Then, these components are stored in a sequence of image vectors and finally used as input to the CNN.

 ISBN 978-80-86943-42-8

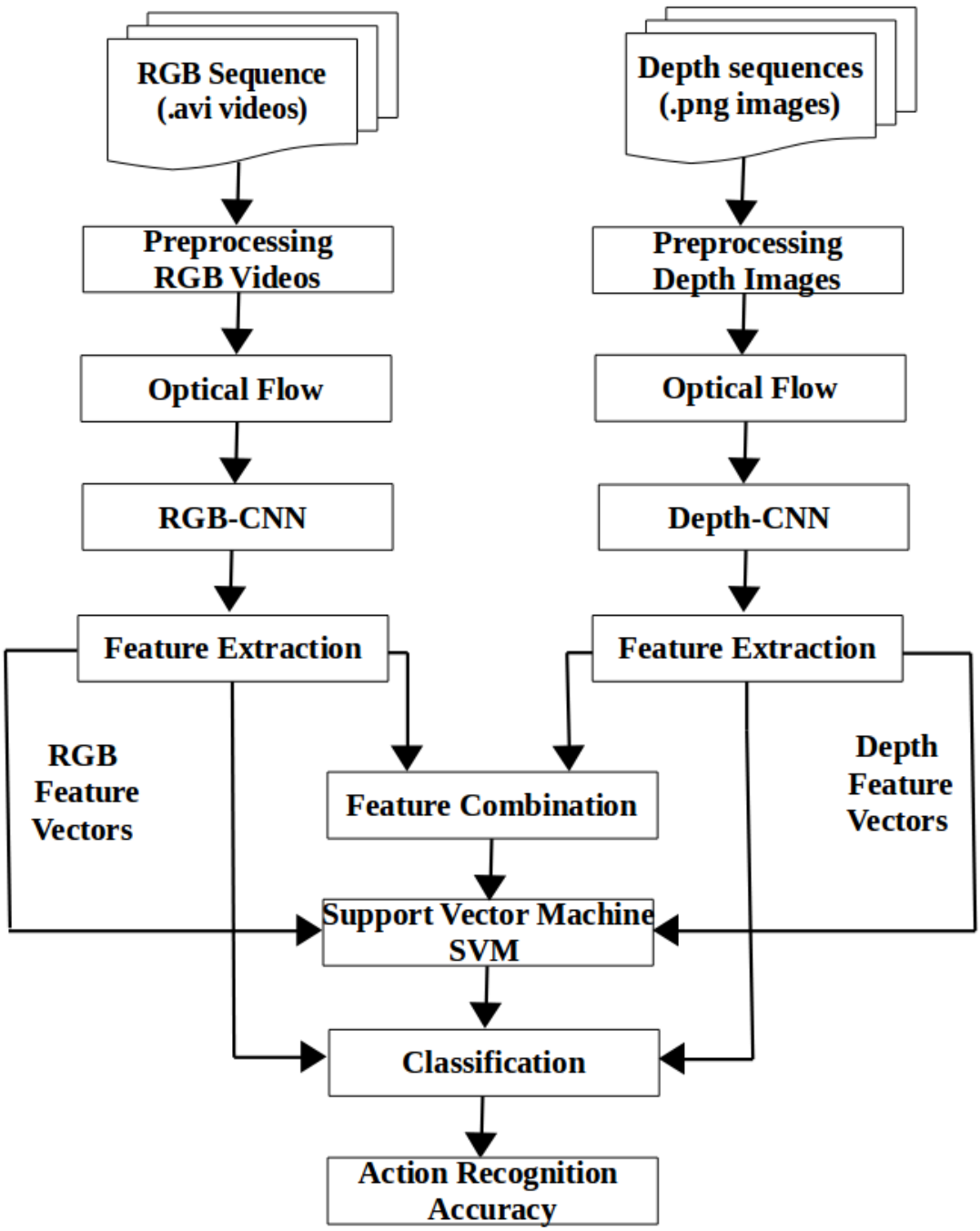


Figure 1: General convolutional Neural networks structure for human action recognition form RGB and Depth video action

3.2.2 Convolutional Neural Networks

The Convolutional Neural Networks (CNN) are representing a hierarchical architecture that can be trained to perform various detection, classification and recognition tasks. A standard CNN consists of two essential components: a feature extractor and a classifier. The feature extractor is used to filter input images into feature maps which are represented a set of features from the images. These features are represented a low-dimensional vector and include corners, lines, edges, etc., which are relatively invariant to position shifting or distortions [CS17]. Then the output from the feature extractor is fed into the classifier, which is usually based on traditional artificial neural networks. In this work, the 3D CNN task involves not only tracking the temporal movement information but also extraction of spatial features. Feature Extraction from 3D CNN describes the process of utilizing the network weights and architecture to fit a new problem. For this purpose, data similarity and data size has to be taken into account.

ISBN 978-80-86943-42-8

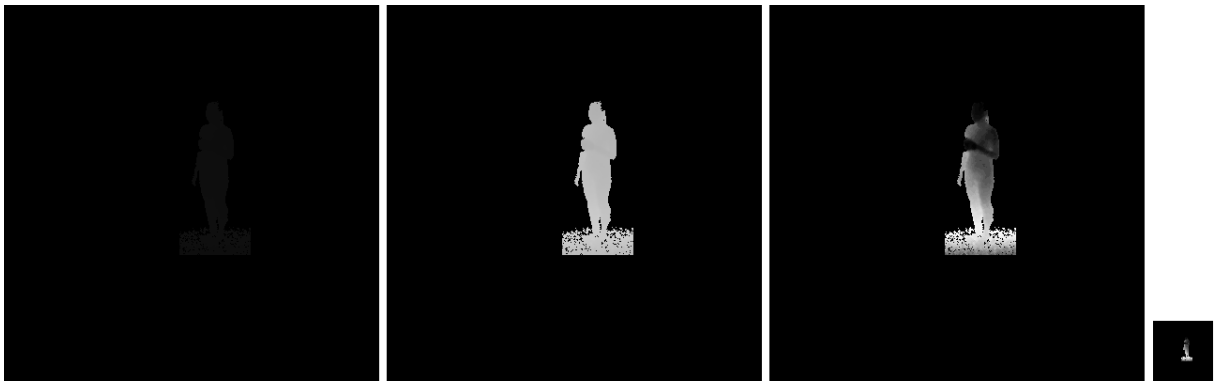


Figure 2: Sample frames of action drinking, from left to right: original depth map, depth map rescaled to range [0,255], rescaled depth map after histogram spreading, final depth map resized to 64x64 pixels.

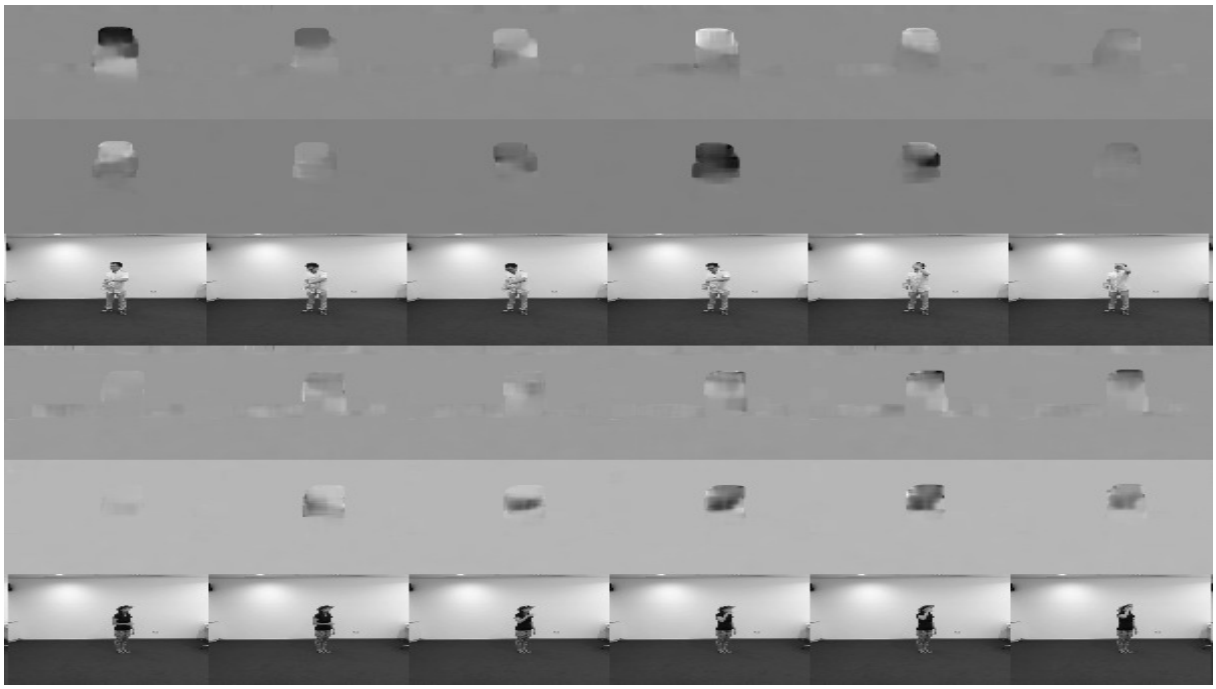


Figure 3: From top: optical flow displacement dx, dy, original gray-scale frame, these frames are extracted in the middle of optical flow volume and corresponding RGB frames, action eating (top) and drinking (bottom)

This CNN system can be trained directly for the task of motion prediction in terms of optical flow for the task of human action recognition. The OF-CNN model which represented in Figure 4, that consists of 4 convolution layers each followed by a max-pooling layer and relu activation. Two fully-connected layers and softmax activation generated the prediction output. In this task, the input takes the optical flow vectors of a single optical flow image volume and treats the individual frames as image channels, and the size of convolution filters is adjusted from 5 to 3. This CNN system can be trained in two different fold, either evaluated directly OF-CNN or by depending on the feature extracted from OF-CNN which are evaluated by using multi-class SVM classifier.

4 EXPERIMENT AND RESULTS

To improve our method and because of CNN require large datasets for training and testing purposes, the Nanyang Technological University's Red Blue Green and Depth information (NTU RGB+D) datasets [SLNW16] is used in order to provide reliable results and the accuracy of the system. The NTU RGB+D is one of the largest scale benchmark dataset for 3D action recognition. It provided 56880 RGB+D video samples of 60 distinct actions. The 60 action classes in NTU RGB+D dataset are presented as: *"drinking, eating, brushing teeth, brushing hair, dropping, picking up, throwing, sitting down, standing up, clapping, reading, writing, tearing up paper, wearing jacket, taking off jacket, wearing a shoe, taking off a shoe, wearing on glasses, taking off glasses, putting on*

ISBN 978-80-86943-42-8

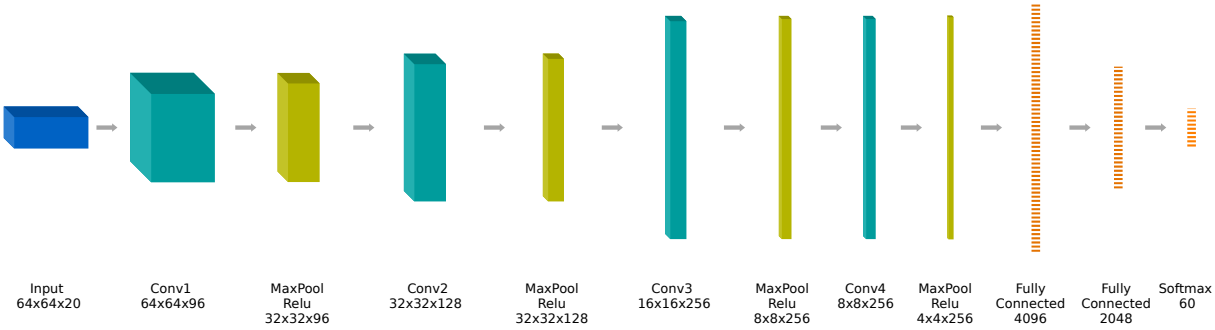


Figure 4: OF-CNN architecture, transformation of input volume, convolutional, pooling and relu layer and softmax output

a hat/cap, taking off a hat/cap, cheering up, hand waving, kicking something, reaching into self pocket, hopping, jumping up, making/answering a phone call, playing with phone, typing, pointing to something, taking selfie, checking time on watch, rubbing two hands together, bowing, shaking head, wiping face, saluting, putting palms together, crossing hands in front, sneezing/coughing, staggering, falling down, headache, touching chest, touching back, touching neck, vomiting, fanning self, punching/slapping other person, kicking other person, pushing other person, patting other's back, pointing to the other person, hugging, giving something to other person, touching other person's pocket, handshaking, walking towards each other, and walking apart from each other". All tested results are done on an Intel Pentium G4600 at 3.6GHz and 8GB RAM, for training a single class of NTU60 model takes 1650s on average, respectively 44 hours (without data loading). See Figure 5 which is showed different action from RGB channel. In this work, only RGB data (136 GB) and masked depth maps (83 GB) are considered. Two different train-test splits for the NTU dataset are proposed. A cross subject split divides the dataset into two groups each containing 20 distinct subjects with 40,320 training and 16,560 test samples, respectively 71% and 29%. The cross view split utilizes the different camera views for each action. The training set contains 37,920 samples (66,6%) with front and side views of the action and the test set holds 18,960 samples (33,3%) with a 45-degree view. Due to time constraints the OF-RGB-depth-CNN, the cross subject split is used in our experimentation results. Our **case study** of this system is illustrated in the next three steps:

- Optical flow is computed from RGB video data, by reducing a single video sequence to 10 optical flow images (OF-volume), this resulting in 20 channel as an input from each video sequence from vertical and horizontal components (d_x, d_y), the image width and height is reduced to 64 pixels. These optical flow volumes are expected to hold temporal motion information suitable for the action recognition task, and they are fed to CNN model for feature extraction and training.

- Optical flow is computed by using depth data from corresponding actions of the same RGB dataset is trained for comparison. In contrast to the OF-RGB-CNN approach, a depth data volume utilizes 10 frames with equal distance which are extracted from the full sequence of depth images. Further, the OF-depth-CNN is trained and evaluated in a similar manner of previous RGB-CNN.

- Both CNN models (OF-RGB-Depth-CNN) are used to explore the possibilities of feature extraction in combination with support vector machine classification. For this purpose, each CNN serves as a fixed feature extractor. The evaluation of this classification method is then done separately for each CNN as well as for a fused model combining feature vectors of both CNN. To explore another common method which can yield further accuracy improvement, the SVM is trained combining feature vectors extracted from the OF-RGB-CNN and OF-depth-CNN models. The larger feature vectors are expected to deliver improved performance. For this purpose the feature vectors are joined by concatenation, to form a single feature vector of dimension 3072 from each video frames. Feature vectors are not further processed, e.g. normalized and rescaled. The multi-class SVM with RBF Kernel is used and setup for training and test procedure [AaP18].

The comparison results are presented in Table 1, which is compared the previous case study results based on three different models from different input channels

ISBN 978-80-86943-42-8

Figure 5: NTU dataset images example [SNGW18]

(RGB, depth, RGBD). From this results, we demonstrate that a CNN with low prediction accuracy can give feature values that yield better classification results with SVM.

Table 2, shows the comparison results with the other state-of-the-art methods using the same datasets.

Model	Modality	Accuracy
OF-RGB-CNN	RGB	40,6%
OF-RGB-CNN-SVM	RGB	**44**%
OF-Depth-CNN	Depth	46,9%
OF-Depth-CNN-SVM	Depth	**50,2**%
OF-RGB-Depth-CNN-SVM	RGB+Depth	**65**%

Table 1: accuracy comparison of all CNN and SVM approaches

Method	Modality	Accuracy
HOG^2 [SNGW18]	Depth	32.24%
Super Normal Vector [SNGW18]	Depth	31.82%
HON4D [SNGW18]	Depth	30.56%
LSTM Encoder-Decoder [Luo17]	RGB	56%
OF-RGBD-CNN- SVM (our)	**RGB+Depth**	**65**%

Table 2: Comparison with the state-of-the-art methods on NTU-RGBD cross subject split dataset

5 CONCLUSION

This paper has presented a deep 3D convolutional neural network based model for classifying and recognizing human actions based on RGB-D data. These models extract features from both spatio and temporal dimensions by performing 3D CNN. The experimental results on NTU RGB+D datasets demonstrate that fusion of different modalities can give better performance than using each modality individually, which mean that the incorporation of RGB and depth modalities to compute 3D CNN feature vectors and supervised learning for the evaluation that yields better prediction accuracy compared to the original CNN. In this work, a support vector machine classifier is used and the accuracy results values outperform the results from baseline CNN in the individual modalities. Training a 3D CNN which provides reliable results and requires not only large datasets but also time, hardware and knowledge. Especially the impact of dataset size is crucial to CNN applications. For Future, We will explore the unsupervised training of 3D CNN models.

ACKNOWLEDGEMENTS

This work was partially supported by Ministry of Higher Education and Scientific Research (MHESR), Iraq, and University of Koblenz-Landau, Germany.

6 REFERENCES

[AaP18] Rawya Al-akam and Dietrich Paulus. Local Feature Extraction from RGB and

ISBN 978-80-86943-42-8

Depth Videos for Human Action Recognition. *International Journal of Machine Learning and Computing (IJMLC)*, 8(3):274–279, 2018.

[AEV17] Ana Paula G S De Almeida, Bruno Luiggi M Espinoza, and Flavio De Barros Vidal. Human action recognition in videos: A comparative evaluation of the classical and velocity adaptation space-time interest points techniques. In *Conference on Computer Graphics, Visualization and Computer Vision WSCG*, 2017.

[Ath15] Keegan Athiwaratkun, Ben Kang. Feature Representation in Convolutional Neural Networks. *arXiv1507.02313 [cs]*, pages 6–11, 2015.

[CDF+04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual Categorization with Bags of Keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[CS17] Jing Chang and Jin Sha. An efficient implementation of 2d convolution in cnn. *IEICE Electronics Express*, 14(1):1–8, 01 2017.

[Far03] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, SCIA'03, pages 363–370, Berlin, Heidelberg, 2003. Springer-Verlag.

[HSXS13] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013.

[JBCS13] Yu Gang Jiang, Subhabrata Bhattacharya, Shih Fu Chang, and Mubarak Shah. High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval*, 2(2):73–101, 2013.

[Lap05] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[LSR08] Ivan Laptev, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. *Computer Vision and Pattern Recognition, CVPR 2008*, pages 0–7, 2008.

[Luo17] Boya Huang De-An Alahi Alexandre Fei-Fei Li Luo, Zelun Peng. Unsupervised Learning of Long-Term Motion Dynamics for Videos. *CVPR 2017, Computer Vision and Pattern Recognition (cs.CV)*, 2017.

[PD07] F Perronnin and C Dance. Fisher Kenrels on Visual Vocabularies for Image Categorizaton. *IEEE Computer Vision and Pattern Recognition, CVPR '07.*, 2007.

[PW14] Zhimin Gao-Jing Zhang Chang Tang-Philip O. Ogunbona Pichao Wang, Wanqing Li. Action Recognition From Depth Maps Using Deep Convolutional Neural Networks. *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS*, 46(4):498–509, 2014.

[RASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519, 2014.

[RF16] M. Radolko and F. Farhadifard. Using trajectories derived by dense optical flows as a spatial component in background subtraction. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG*, 2016.

[SHJ17] Xinhang Song, Luis Herranz, and Shuqiang Jiang. Depth CNNs for RGB-D Scene Recognition: Learning From Scratch Better Than Transferring From RGB-CNNs. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.

[SLNW16] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[SLY16] Ling Shao, Li Liu, and Mengyang Yu. Kernelized Multiview Projection for Robust Action Recognition. *International Journal of Computer Vision*, 118(2):115–129, 2016.

[SNGW18] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 40(5), 2018.

[SZ14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*,

abs/1406.2199, 2014.

[VVV16] Kushal Vyas, Yash Vora, and Raj Vastani. Using Bag of Visual Words and Spatial Pyramid Matching for Object Classification Along with Applications for RIS. *Procedia Computer Science*, 89:457–464, 2016.

[Wan14] Xiaolong Lin Liang Wang-Meng Zuo Wangmeng Wang, Keze. 3d human activity recognition with reconfigurable convolutional neural networks. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 97–106, New York, NY, USA, 2014. ACM.

[WS13] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558, 2013.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27 (Proceedings of NIPS)*, 27:1–9, 2014.

[YCXL17] Sheng Yu, Yun Cheng, Li Xie, and Shao-Zi Li. Fully convolutional networks for action recognition. *IET Computer Vision*, 11(8):744–749, 2017.

 ISBN 978-80-86943-42-8

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

Realistic Lens Distortion Rendering

Martin Lambers
Computer Graphics Group
University of Siegen
Hoelderlinstrasse 3
57076 Siegen
martin.lambers@
uni-siegen.de

Hendrik Sommerhoff
Computer Graphics Group
University of Siegen
Hoelderlinstrasse 3
57076 Siegen
hendrik.sommerhoff@
student.uni-siegen.de

Andreas Kolb
Computer Graphics Group
University of Siegen
Hoelderlinstrasse 3
57076 Siegen
andreas.kolb@
uni-siegen.de

ABSTRACT

Rendering images with lens distortion that matches real cameras requires a camera model that allows calibration of relevant parameters based on real imagery. This requirement is not fulfilled for camera models typically used in the field of Computer Graphics.

In this paper, we present two approaches to integrate realistic lens distortions effects into any graphics pipeline. Both approaches are based on the most widely used camera model in Computer Vision, and thus can reproduce the behavior of real calibrated cameras.

The advantages and drawbacks of the two approaches are compared, and both are verified by recovering rendering parameters through a calibration performed on rendered images.

Keywords

Lens distortion, Camera calibration, Camera model, OpenCV

1 INTRODUCTION

In Computer Graphics, the prevalent camera model is the pinhole camera model, which is free of distortions and other detrimental effects. Real world cameras, on the other hand, use lens systems that lead to a variety of effects not covered by the pinhole model, including depth of field, chromatic aberration, and distortions. This paper focusses on the latter.

In Computer Vision, distortions must be taken into account during 3D scene analysis. A variety of camera models have been suggested to model the relevant effects; Sturm et al. [1] give an overview. The dominant model in practical use is a polynomial model based on the work of Heikkilä [2, 3] and Zhang [4] and is implemented in the most widely used Computer Vision software packages: OpenCV [5] and Matlab/Simulink [6]. In the following, we refer to this camera model as the standard model. Typical Computer Vision applications estimate the distortion parameters of the standard model for their camera system in a calibration step, and then undistort the input images accordingly before using them in further processing stages.

For a variety of applications, including analysis-by-synthesis techniques [7], sensor simulation [8], and special effects in films [9], it is useful to apply the reverse process, i.e. to synthesize images that exhibit realistic distortions by applying a camera model. Using the standard model for this purpose has the advantage that model parameters of existing calibrated cameras can be used directly, with immediate practical benefit to all application areas mentioned above.

In this paper, we present and compare two ways of integrating realistic distortions based on the standard camera model into graphics pipelines. One is based on preprocessing the geometry, and the other is based on postprocessing generated images. We show that both methods have unique advantages and limitations, and the choice of method therefore depends on the application. We verify both approaches by showing that standard model calibration applied to synthesized images recovers the distortion parameters with high accuracy.

2 RELATED WORK

In Computer Graphics, camera models that are more realistic than the pinhole model are typically based on a geometric description of the lens system that is then integrated into ray tracing pipelines [10, 11]. This approach is of limited use if the goal is to render images that match the characteristics of an existing camera, as suitable parameters cannot be derived automatically. Furthermore, this approach excludes rasterization pipelines, which is problematic for applications that benefit from fast image generation.

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.4

In contrast, using a Computer Vision camera model allows to apply parameters obtained by calibrating a real camera and, as shown in Sec. 3, can be done in any graphics pipeline.

Sturm et al. [1] give an overview of camera models in Computer Vision. Most models account for radial distortion (e.g. barrel and pincushion distortion, caused by stronger bending of light rays near the edges of a lens than at its optical center) and tangential distortion (caused by imperfect parallelism between lens and image plane). Some also account for thin prism distortion (caused by a slightly decentered lens, modeled via an oriented thin prism in front of a perfectly centered lens), and tilted sensor distortion (caused by a rotation of the image plane around the optical axis).

The complete formulas for the standard model [5] compute distorted pixel coordinates from undistorted pixel coordinates and use parameters $k_1,\ldots,k_6$ for radial distortion, $p_1\,p_2$ for tangential distortion, $s_1,\ldots,s_4$ for thin prism distortion, and τ_1,τ_2 for tilted sensor distortion.

In practice, thin prism distortion and tilted sensor distortion are usually ignored, and radial distortion is limited to two or at maximum three parameters (the others are assumed to be zero). This is documented by the fact that the calibration functions of OpenCV[1] and Matlab/Simulink[2] estimate only the parameters k_1,k_2,p_1,p_2 and optionally k_3 by default.

In the following, we focus on the standard camera model of Computer Vision, and apply it to arbitrary rendering pipelines via either geometry preprocessing or image postprocessing.

3 METHOD

We first summarize the standard model in Sec. 3.1, focussing on the aspects relevant for this paper and incorporating its intrinsic camera parameters into the projection matrix of a pinhole camera model. On this basis, simulating lens distortion can be done in one of two ways:

- By *preprocessing geometry*. In this approach, each vertex of the input geometry is manipulated such that its position in image space after rendering corresponds to a distorted image.
- By *postprocessing images*. In this approach, an undistorted image is rendered based on the pinhole camera model, and distorted in a postprocessing step based on the standard model.

These approaches are described in detail in the Sec. 3.2 and Sec. 3.3.

[1] https://docs.opencv.org/3.4.0/dc/dbb/tutorial_py_calibration.html

[2] https://mathworks.com/help/vision/ug/camera-calibration.html

```
vec4 clipCoord = P * position;
vec2 ndcCoord = clipCoord.xy / clipCoord.w;
vec2 pixelCoord = vec2(
    (ndcCoord.x * 0.5 + 0.5) * w,
    (0.5 - ndcCoord.y * 0.5) * h);
// apply the standard model to pixelCoord
ndcCoord.x = (pixelCoord.x / w) * 2.0 - 1.0;
ndcCoord.y = 1.0 - (pixelCoord.y / h) * 2.0;
clipCoord.xy = ndcCoord * clipCoord.w;
```

Algorithm 1: GLSL code fragment for applying the standard model in the vertex shader.

3.1 The Standard Model

The standard model, reduced to the part that is relevant in this discussion, has the following parameters: the camera intrinsic parameters, consisting of the principal point c_x,c_y and the focal lengths f_x,f_y (both in pixel units), the radial distortion parameters k_1,k_2, and the tangential distortion parameters p_1,p_2. The model computes distorted pixel coordinates u,v from undistorted pixel coordinates x,y by first computing normalized image coordinates s,t with distance r to the principal point, applying the distortion, and then reverting the normalization [5]:

$$\begin{aligned} s &= \frac{x-c_x}{f_x} \\ t &= \frac{y-c_y}{f_y} \\ r^2 &= s^2+t^2 \\ d &= 1+k_1r^2+k_2r^4 \\ u &= (sd+(2p_1st+p_2(r^2+2s^2)))f_x+c_x \\ v &= (td+(p_1(r^2+2t^2)+2p_2st))f_y+c_y \end{aligned} \tag{1}$$

Here, the undistorted pixel coordinates x,y are equivalent to pixel coordinates generated with the pinhole camera model of a standard graphics pipeline when the camera intrinsic parameters c_x,c_y,f_x,f_y are accounted for in the projection matrix. This matrix is typically defined by a viewing frustum given by the clipping plane coordinates l,r,b,t for the left, right, bottom, and top plane. These values have to be multiplied by the near plane value n; here we assume $n=1$ for simplicity. Given the image size $w\times h$, suitable clipping plane coordinates can be computed from the camera intrinsic parameters as follows:

$$\begin{aligned} l &= -\frac{c_x+0.5}{f_x} \\ r &= \frac{w}{f_x}+l \\ b &= -\frac{c_y+0.5}{f_y} \\ t &= \frac{h}{f_y}+b \end{aligned}$$

ISBN 978-80-86943-42-8

Using this frustum to define the projection matrix in a standard graphics pipeline accounts for the camera intrinsic parameters of the standard model. The remaining problem is to integrate the lens distortion parameters k_1, k_2, p_1, p_2. This is discussed in the following sections.

3.2 Preprocessing Geometry

In this approach, each input vertex is manipulated such that its image space coordinates match the distorted coordinates of the standard model.

In a standard graphics pipeline, this manipulation is typically done in the vertex shader. Since the standard model operates on pixel coordinates, we first apply the projection matrix from Sec. 3.1 to each vertex, resulting in clip coordinates, and then divide by the homogeneous coordinate to get normalized device coordinates (NDC). By applying the viewport transformation, these are transformed to window coordinates, which are equivalent to pixel coordinates in the standard model. After modifying the x and y components of the window coordinates to account for lens distortion according to Eq. 1, we transform back to clip coordinates. See Alg. 1 for an OpenGL vertex shader code fragment.

This approach has two limitations.

First, modifying clip coordinates in this way means that a fundamental assumption of the graphics pipeline, namely that straight lines in model space map to straight lines in image space, is no longer fulfilled. This leads to errors. A similar problem occurs in graphics applications that project onto non-planar surfaces, e.g. shadow maps [12] and dynamic environment maps [13] that aim to reduce memory usage. There, the errors are considered acceptable if the tessellation of the input geometry is fine enough such that triangle edges in image space are short. Whether this condition is met in our case depends on the application.

Second, our vertex modification takes place before clipping, and therefore includes vertices that lie outside the domain of the standard model. Depending on the distortion parameters, transforming these vertices may place them into image space, resulting in invalid triangles that ruin the rendering result. To avoid this problem, we discard triangles that contain at least one vertex outside of the view frustum. A tolerance parameter δ can be applied during this test to avoid holes in the final image caused by triangles that are partly inside the frustum: a vertex is discarded if its unmodified NDC xy coordinates lie outside $[-1-\delta, 1+\delta]^2$. Since the preprocessing approach requires a finely detailed geometry anyway, simply using $\delta = 0.1$ should work fine. We used this value for all of our tests.

For certain types of distortion, mainly barrel distortion (see Fig. 1), we must additionally account for vertices that lie outside of the pinhole camera frustum but may be mapped into image space nonetheless. This is done by adding a distortion-dependent value D to the parameter δ. Given the inverse of the standard model (see Sec. 3.3 for details), we can determine a lower bound for D automatically by undistorting the distorted image space corner coordinates $(0,0), (w,0), (w,h), (0,h)$, transforming them to NDC coordinates, and setting D to the maximum of the absolute value of each coordinate, minus one.

3.3 Postprocessing Images

In this approach, the scene is first rendered into an undistorted image using an unmodified graphics pipeline based on a pinhole camera with the projection matrix from Sec. 3.1. The result is then transformed into a distorted image by applying the standard model in a postprocessing step, e.g. using a fragment shader.

This postprocessing step requires the computation of undistorted pixel coordinates (x,y) from distorted pixel coordinates (u,v), i.e. the inverse of Eq. 1. This inversion is not a trivial problem; several approaches exist, but none supports the full set of parameters of the original standard model. For example, Drap and Lefèvre propose an exact inversion, but for radial distortion only [14].

We apply ideas by Heikkilä [3] to invert Eq. 1 using an approximation based on Taylor series. Note that his camera model differs from the standard model; in particular, it computes undistorted pixel coordinates from distorted pixel coordinates. Nevertheless, his inversion process is still applicable. The resulting formulas support radial distortion parameters k_1, k_2 and tangential distortion parameters p_1, p_2, which is sufficient in practice:

$$\begin{aligned} s &= \frac{u - c_x}{f_x} \\ t &= \frac{v - c_y}{f_y} \\ r^2 &= s^2 + t^2 \\ d_1 &= k_1 r^2 + k_2 r^4 \\ d_2 &= \frac{1}{4k_1 r^2 + 6k_2 r^4 + 8p_1 t + 8p_2 s + 1} \\ x &= (s - d_2(d_1 s + 2p_1 st + p_2(r^2 + 2s^2)))f_x + c_x \\ y &= (t - d_2(d_1 t + p_1(r^2 + 2t^2) + 2p_2 st))f_y + c_y \end{aligned} \tag{2}$$

Note that the postprocessing step can only fill areas in the distorted image for which information exists in the undistorted image. For certain types of distortion, mainly barrel distortion (see Fig. 1), this means that some areas of the result remain unfilled. This can only be alleviated by using both an enlarged frustum and an increased resolution when rendering the undistorted

 ISBN 978-80-86943-42-8

	Preprocessing geometry	Postprocessing images
Distortion model completeness	full	limited to radial and tangential
Prerequisites	finely detailed geometry	none
Result completeness	full	may have unfilled areas
Rendered data types	all	limited to interpolatable, relocatable data
Complexity	geometry-dependent	resolution-dependent

Table 1: Comparison of the pre- and postprocessing approaches to lens distortion rendering based on the standard model. See Sec. 3.4 for details.

image. Note that while it is possible to derive suitable frustum and resolution parameters by computing undistorted coordinates for the distorted image space corner coordinates $(0,0),(w,0),(w,h),(0,h)$, similar to method described for the preprocessing approach, we did not do so in our tests for simplicity.

3.4 Discussion

In this section, we discuss several aspects of the preprocessing and postprocessing approaches, summarized in Tab. 1.

Distortion model completeness: In the preprocessing approach, we apply the forward standard model and thus can use the full formulas unchanged, i.e. with support for all parameters, including thin prism and tilted sensor distortion if relevant. The postprocessing approach requires the inverse model, and no inversion is known that accounts for all parameters. It is therefore limited to radial and tangential distortion with parameters k_1,k_2,p_1,p_2, but this should be sufficient for the majority of applications.

Prerequisites: Applying the preprocessing approach requires finely tessellated geometry to keep errors small. Not all applications may be able to make such guarantees. The postprocessing approach does not have this limitation.

Result completeness: While the preprocessing approach can map geometry outside of the pinhole camera view frustum into the distorted image, such information is not available to the postprocessing approach unless an enlarged frustum and increased resolution are used for the undistorted image. See Fig. 1.

Rendered data types: The postprocessing approach will usually map undistorted pixels with interpolation to the distorted image. This is fine e.g. for RGB images, but may break for other kinds of data that special applications may render into images, e.g. object IDs or 2D pixel flow. In these cases, only the preprocessing approach can be applied.

Computational complexity: The complexity of the postprocessing approach depends on the number of vertices in the input geometry, while the complexity of the postprocessing approach depends on the number of output pixels. While the preprocessing approach can be integrated directly into any pipeline, the postprocessing approach requires an additional render pass.

4 RESULTS

We implemented both the preprocessing and the postprocessing approach in a standard OpenGL rendering pipeline. To verify that our implementation produces results that match the OpenCV/Matlab implementation of the standard model, we varied the model parameters $c_x,c_y,f_x,f_y,k_1,k_2,p_1,p_2$, then rendered a set of 17 images of size 800×600 for each parameter set, containing the standard OpenCV checkerboard calibration pattern in various 3D positions and orientations, and then used the OpenCV `calibrate.py` script to estimate the model parameters from the rendered images. Note that OpenCV also supports a circle grid calibration pattern, but we chose to use the more widely used checkerboard pattern.

In most cases, the original parameters were recovered with high accuracy, even though the rendered set of images was of low quality for calibration purposes. The average recovery error was less than $1\,\%$ for $c_x,c_y,f_x,f_y,k_2,p_1,p_2$. Interestingly, for k_1 the error was significantly larger, however this did not cause noticeable errors in the undistorted images that were produced for verification purposes. For a few sets, calibration failed, mostly caused by parts of the checkerboard pattern not being visible in some images.

Fig. 2 shows a visual verification: first, an undistorted image is rendered, then a distorted one with a specific set of parameters, and this distorted image is finally undistorted using OpenCV with the same parameters. The first an last image show only minimal differences.

5 CONCLUSION

We presented two methods to accurately render images that match the characteristics of real cameras regarding implicit parameters and lens distortion. Both methods are based on the most widely used camera model in Computer Vision, and can be integrated into any rendering pipeline.

We highlighted the specific advantages and drawbacks of each approach to help implementers pick the right approach for a given application.

 ISBN 978-80-86943-42-8

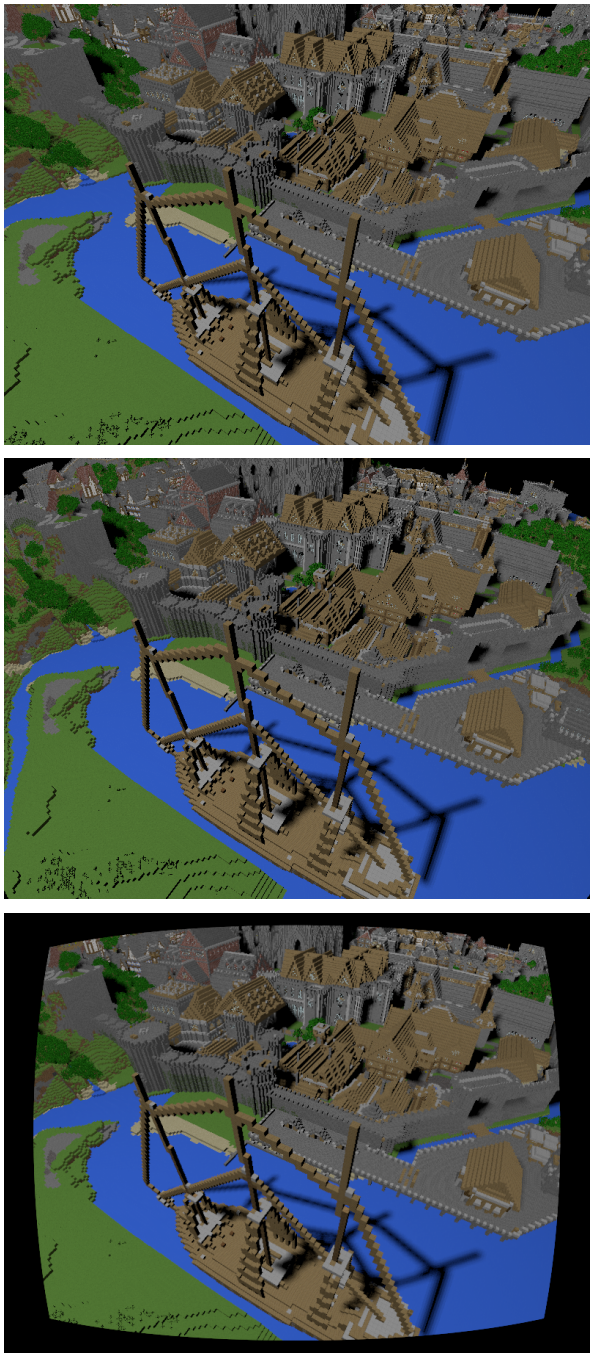


Figure 1: Effects of barrel distortion ($k_1 = -0.11, k_2 = 0, p_1 = 0, p_2 = 0$). From top to bottom: undistorted image, distorted image from preprocessing geometry, and distorted image from postprocessing the undistorted image.

Figure 2: From top to bottom: undistorted image of size $800, 600$ rendered with intrinsic parameters $c_x = 399.5$, $c_y = 299.5$, $f_x = f_y = 400$, distorted image rendered with parameters $k_1 = -0.05$, $k_2 = 0.01$, $p_1 = 0.03$, $p_2 = -0.01$, and undistorted image produced from the distorted image by OpenCV using the same parameters.

ISBN 978-80-86943-42-8

6 ACKNOWLEDGMENTS

The work is partially funded by the German Research Foundation (DFG), grants Ko-2960-12/1 and Ko-2960-13/1.

The scene displayed in Fig. 1 and Fig. 2 is the Rungholt scene from McGuire graphics data [15].

7 REFERENCES

[1] P. Sturm, S. Ramalingam, S. Gasparini, and J. Barreto. *Camera Models and Fundamental Concepts Used in Geometric Computer Vision*. Now Foundations and Trends, 2011.

[2] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, pages 1106–1112, Jun 1997.

[3] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, Oct 2000.

[4] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[5] OpenCV contributors. OpenCV camera model description. `https://docs.opencv.org/3.4.0/da/d54/group__imgproc__transform.html#ga7dfb72c9cf9780a347fbe3d1c47e5d5a`. Accessed 2018-03-14.

[6] Mathworks. Matlab / Simulink camera parameters. `https://www.mathworks.com/help/vision/ref/cameraparameters.html`. Accessed 2018-03-14.

[7] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D object pose estimation using 3D object coordinates. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 536–551, 2014.

[8] M. Lambers, S. Hoberg, and A. Kolb. Simulation of time-of-flight sensors for evaluation of chip layout variants. *IEEE Sensors Journal*, 15(7):4019–4026, July 2015.

[9] D. Roble. Vision in film and special effects. *ACM SIGGRAPH Comput. Graph. Newsletter*, 33(4):58–60, November 1999.

[10] C. Kolb, D. Mitchell, and P. Hanrahan. A realistic camera model for computer graphics. In *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 317–324, 1995.

[11] J. Wu, C. Zheng, X. Hu, and C. Li. An accurate and practical camera lens model for rendering realistic lens effects. In *Int. Conf. on Computer-Aided Design and Computer Graphics*, pages 63–70, September 2011.

[12] D. Scherzer, M. Wimmer, and W. Purgathofer. A survey of real-time hard shadow mapping methods. *Computer Graphics Forum*, 30(1):169–186, 2011.

[13] T. Y. Ho, L. Wan, C. S. Leung, P. M. Lam, and T. T. Wong. Unicube for dynamic environment mapping. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 17(1):51–63, Jan 2011.

[14] P. Drap and J. Lefèvre. An exact formula for calculating inverse radial lens distortions. *MDPI Sensors*, 16(6), 2016.

[15] Morgan McGuire. Computer graphics archive. `https://casual-effects.com/data`, July 2017.

 ISBN 978-80-86943-42-8

Time Series Social Network Visualization Based on Dimension Reduction

Maha Al-Ghalibi
Computational Visualistics
University of Koblenz-Landau
Universitätsstr.1
56070 Koblenz,
Germany.
alghalibi@uni-koblenz.de

Adil Al-Azzawi
Electrical Engineering and Computer Science, University of Missouri-Columbia
USA (65211), MO, Columbia
aaadn5@mail.missouri.edu

ABSTRACT

Social networks are in general dynamically due to the involvement of many people on the web such as Facebook, Twitter, and Snapchat, etc. The meaningful visualization and analysis of social network is challenging due to its dynamic nature, the mobility of nodes in the network and extremely large size. In this paper, we consider the higher dimensionality issue of social networks regarding time series social network construction and visualization. To solve this issue, we develop a statically data-mining based approach for dimensionality reduction in social networks. Basically, we find that each sub-social network's model has different dimensions by nodes and links which are sampled originally from an *m*-dimensional metric space. Experimentally, we find that the *m*-dimensional features for each sub-network cause fail connections in time-series during the network reconstruction model for visualization. Therefore, we propose a new dimension reduction approach that is based on developing an SVD algorithm by relying on select significant sub features. Then we extract time features from the feature space of the original dataset to visualize the network in a deferent time interval. However, to monitor the network development and also the dimensionality reduction of features help us to speed up the computation time of the shortest path. The social circle Facebook dataset form Stanford is used with its corresponding attributes. The dataset includes node features (profile), circles, and ego networks. The obtained result shows better performances regarding the computation time and network visualization. Moreover, the experimental results show that the proposed system is much faster than the approach based on the whole feature space for closeness centrality computing.

Keywords

Network Visualization, SVD, Mutual Information, Dimensionality Reduction, Feature Selection.

1. INTRODUCTION

In recent years, the use of social networks has become a robust online communication platform supporting millions of users. User behavior in a large-scale data needs visualization and as well as data analysis methods to achieve higher performances [WCG+16]. This paper is focused on clustering to solve the challenging issues of visualization. A dynamic graph discretization and graph clustering used in presenting a hierarchical structure [GSZ+11]. Connected components of are characterized as a Depth First Search (DFS) tree that updates the dynamical changes in graphs. Clustering in dynamic social networks is applied to tolerate any changes that occur in the network. Dynamic changes in the social network are analyzed by monitoring the structural characteristics regarding patterns [Wu10]. This article uses the Top-k Weighted Clustering Coefficient supported maintenance of constructed a graph by performing vertex/edge insertion, vertex/edge deletion and monitoring top-K results [LCZ+17]. Insertion of a new edge or vertex involved without disturbing other vertices and edges. Also, the periodical graph is used that based on betweenness centrality to provide pseudo-polynomial time algorithm [FS15]. Here the edge weights are determined for predicting the shortest path for communication between nodes. For analyzing this type of network, a graphical representation involved in most of the research works. The taxonomy-based graph representation is introduced to analyze, i.e. Network topology, visualization, community detection, [ARK13] the work depends on centrality-based metrics that are computed as eigenvector, closeness centrality, betweenness centrality, page rank, hubs, and authority. Non-centrality-based metrics are reciprocity, transitivity, density, or similarity. Structural similarity-based dynamic network layout visualization is involved in identifying temporal changes in the network [XH16]. A single time slice method is used to determine node distances from which the node's moved distance estimated. The node's best mobile area identified, and then the adjustment is done among the nodes. The currently created network layout compared to the previous layout for predicting similarity to the final layout. Changes in the structure defined by the chang

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.5

centrality metric that enables pairwise comparisons in the evolving network [FPA+12]. They have presented a set of novel metrics for the visual analysis of dynamic networks. Trying to enhance the perception of changes and the gaining of both, overview and detailed insights on the network growth. [CZ17] In this framework, the nodes are matched, and community is assigned. Then the edges of the nodes are updated. In [BJ15] a Time-Varying Graph model for Online Social Networks (TVG-OSN) is presented. An identity map is created at time intervals to analyze the link occurrences among nodes. TVG-OSN supports the determination of user characteristics using different link criteria. A temporal graph model plays a significant role in dynamic social networks. Therefore, the dynamic network environment has been involved with certain challenging issues, and many challenging issues are discussed by many authors that evaluated their performance metrics.

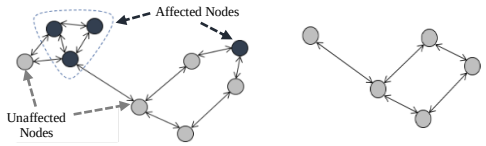
We developed visualization and sentiment analysis approach which first visualize the Facebook dataset in a different time series to show the changes in topology. For time series visualization, the proposed system allows the viewers to visualize the whole data of the Facebook social network in different categories depending on five different time series. We give a statistical visualization of the (total number of nodes, and connections) that has been significantly changed during the growth of the whole social network through the time. After time series visualization, we propose a novel approach for dimensionality reduction called Guided Dimensionality Reduction Approach (GDRA). GDRA solves the problem of dimension reduction by applying a mathematical model to avoid the biased feature selection. We suggest a threshold selection of the significant subfeatures that are ranked by Mutual Information. Moreover, we developed a new version of the Singular Value Description (SVD) by using a multi eigenvalue selection using SVD.

This approach helps our system to reconstruct the sub-network in the time series after removing the non-relevant feature space which causes the NAN connection during the visualization step. Moreover, it potentially reduces the time that has significantly consumed in the short path detection for the visualization of the experimental results. Our last contribution is the construction of a dynamic network model that manages the arrival of the new nodes and departure of nodes that were participating in the network as additional and optional view of our visualization system.

2. BACKGROUND THEORY

Social network analysis and visualization is based on the number of nodes and their connections to the whole entire network. Different nodes connections in a social network can affect the network visualization. Moreover, the fact that affects the decision of the other nodes on drive subject matters. Different influences regular connection node (unaffected nodes and affected nodes) based on their connection types such as direct connection, no connection, or either bi-connection during a specific time in the time series visualization task. This is challenging task in a social network visualization especially the time series task due to the difficulties of reconnect the other sub-network together based on their node connections [WU10].

Time series visualization can cause many issues especially for a complex social network such as the threshold in which the connection nodes can be under the influents decision at that period. Data mining approaches, such as clustering and dimensionality reduction techniques, can be used to model this problem in the social network visualization which is a way of asserting the affected nodes and unaffected nodes [XH16]. For such a visualization task, during time series, there is a decision about whether the influential participant's connection nodes that are either connected or not connected depends mainly on the original connection aspect. This connection is originally constructed based on the number of features that the whole network based on to draw the whole space. In this case, there is a feature vector that does fairly cause the affected influence connection based on the number of NAN this feature vector has. In this case, we assume that the dimension reduction of the feature space may help to solve this kind of complexity. Figure.1 shows an example of the affected and unaffected connections nodes in our situation where in this case, the unaffected node that originally connected to the affected rejoin should be reconstruct again to the unaffected rejoin (nodes).



(a) **(b)**

Figure1: Time series social network visualization issue (a) for the whole network, (b) specific time series

Data Mining Dimensionality Reduction and Feature Selection

The ageneral problem of data mining and machine learning approaches is to handle a high- dimensional data (feature space) due to the huge number of input variables. Dimensionality reduction and feature selection can be made in two ways, By using the feature selection techniques were in this way, keeping the most relevant data (variables) in the original data, or by using the dimensionality reduction techniques where in this case exploiting the redundancy of the input data and by selecting a subset of new data (new variables) [SVM14].

ISBN 978-80-86943-42-8

Singular-Value Decomposition (SVD)

Singular value decomposition (SVD) allows an exact representation of any matrix and makes it easy to eliminate the less important parts of that representation to produce an approximate representation with any desired number of dimensions. Let us assume that X is an $m \times n$ matrix and r is the rank of X. Recall that the rank of a matrix represents the largest number of rows (or equivalently columns) for which a non-zero linear combination of the rows is the all-zero vector 0. The Singular-Value Decomposition (SVD) is described in the Algorithm (1) below [SVM14].

Algorithm: Singular-Value Decomposition

Input: Generate data matrix $\boldsymbol{X}$

Output: New dimensions $\boldsymbol{C}$

1. **Repeat**
2. **Applying** SVD to the matrix X as $X = USV^T$
   $X \rightarrow$ is an $m \times n$ matrix ($m \rightarrow$ no. of vectors)
   $n \rightarrow$ is no. of attributes)
   $U \leftarrow XX^T$matrthe ix of the eigenvectors
   $S \rightarrow$ is a matrix which is diagonal
   $V \leftarrow$ is a matrix of the eigenvectors.
3. **Construct** the covariance matrix from this decomposition by
   $XX^T\ XX^T \leftarrow (USV^T)(USV^T)^T$.
   $V \rightarrow$ orthogonal matrix ($V^TV = I$).
   $XX^T \leftarrow US2U^T$
4. **Compute** the square roots of the eigenvalues of $\boldsymbol{XX^T}$ which are the singular values of X
5. **Until** representing every transaction at vector $x(t)_i$
6. **Return** U^TX
7. **End**

Algorithm1: Singular-value decomposition (SVD)

Mutual Information

Mutual information is a statistical method that measures the relationship between two random variables that simultaneously sampled. It measures how much information has each variable about another. Intuitively, it asks how much each random variable tells us about another one [ZB16]. The formal definition of the mutual information of two random variables X and Y, whose joint distribution is defined by $P(X,Y)$.

Algorithm (2) shows how the MI is computed based on using two random variables [ZB16].

Algorithm: Mutual Information (MI)

Input: Feature Space $\boldsymbol{X}$

Output: Mutual Scoring $\boldsymbol{I}$

1. **Repeat**
2. **Compute** the MI for every two variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ by $\boldsymbol{I(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)p(y)}}$
3. where
   $X \rightarrow$ is the first feature space
   $Y \rightarrow$ is the second feature space
   $P \rightarrow$ is the probability function
4. **Import** the two variables in the feature space
5. **Until** consumed all variable in the whole variables
6. **Return** the mutual scoring vector
7. **End**

Algorithm2: Mutual information (MI).

3. PROPOSED SYSTEM

The pipeline of our proposed system shown in Figure.2

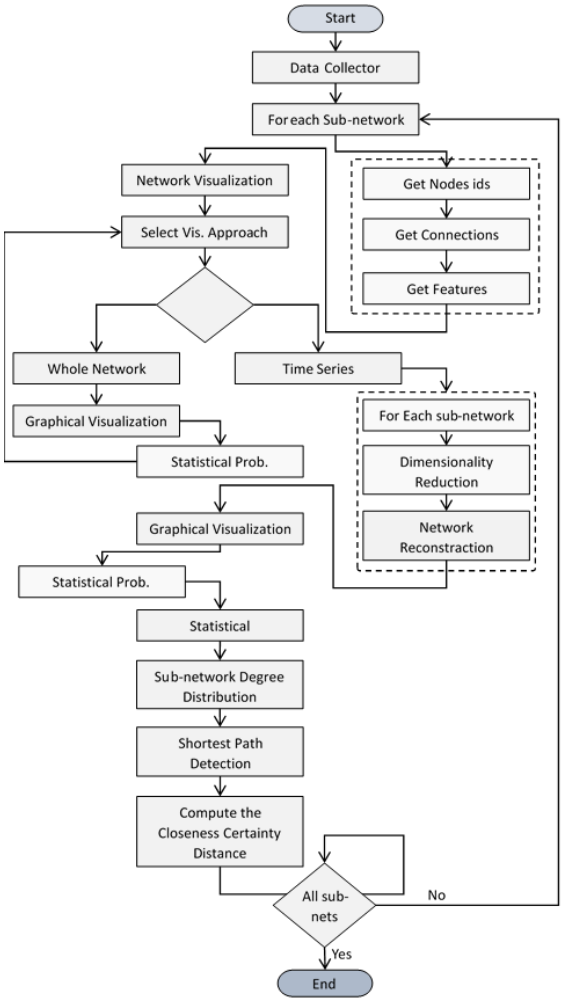


Figure2: Pipeline of Our proposed visualization approach

Any social network model has attributes such as a specific number of nodes (users) and link (connection). The network attributes allow the system to fit the best visualization model based on the dimensional metric space which based on the power-law distribution. Based on the logarithmic relational between the m dimensional scale and the number of users (nodes), we proposed a network visualization model, see Figure 2. It is based on the dimensionality

ISBN 978-80-86943-42-8

reduction approach for social network hypothesis confirming.

The main idea of our model is to reduce the original feature space which is performed in two steps. We using the mutual information (MI) which basically ranks the original features space from 1 to 0. Then, a sub-correlated features set is selected based on the pre-defined threshold. For the feature set selection, we introduce a new algorithm to perform dimensionality reduction (SVD). With this, We select most relevant and correlated sub feature space. The subset used within the model to reduce the unnecessary connection in the social network model according to the NAN connection that has already collected during the network constriction. This is a time-consumption process for whole network dimension, especially for detecting the distances measured by the shortest paths to compute closeness centrality.

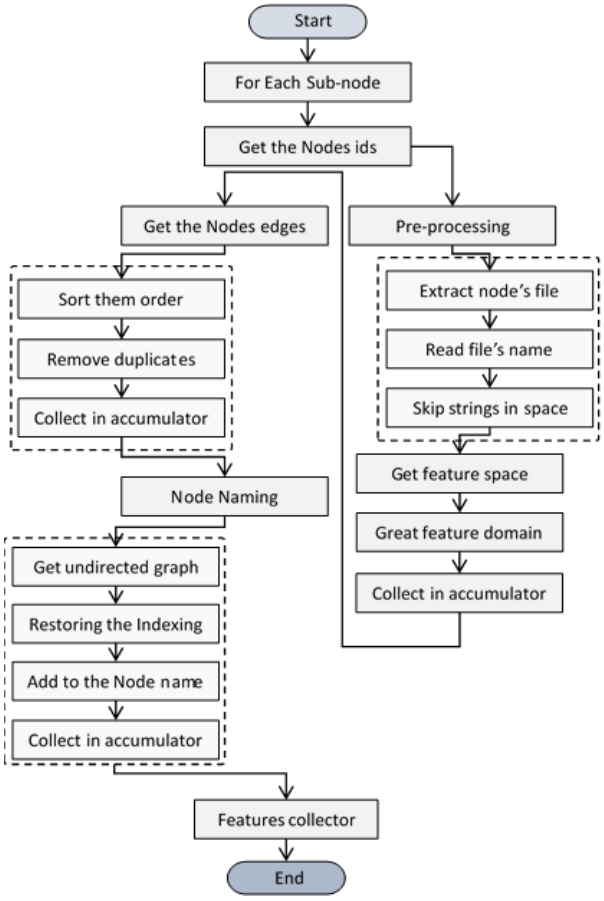


Figure3: Data collector and feature extractor

The visualization of time-series of social networks based on dimensionality reduction consists of two main approaches. The first approach is shown in Figure.3 it comprises the data collection and feature extraction for the whole network including ten subnetworks of the whole Facebook dataset.

The second approach is the visualization of time-series of the network. Figure.4 shows the steps of our approach to select the time series of social networks as well as to visualize whole networks.

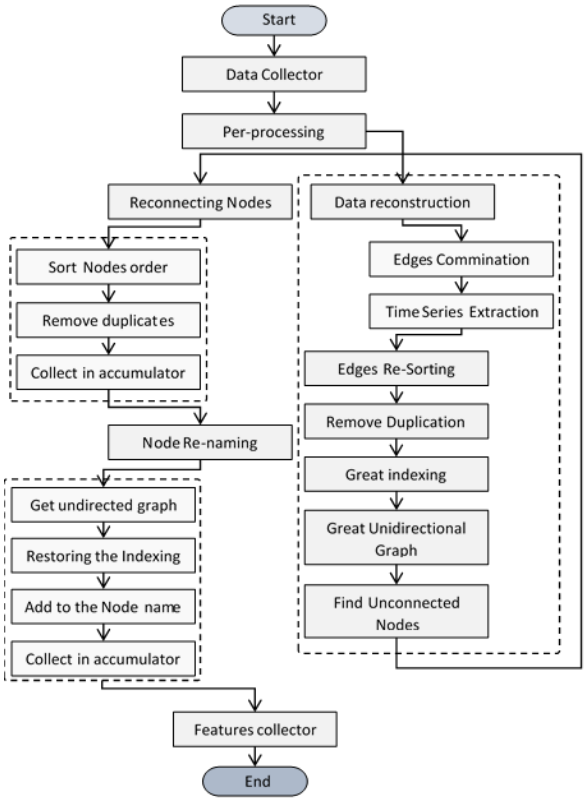


Figure4: Time series network visualization

Sub Feature Selection based Non-Negatively MI.

Our main contribution is a new dimensionality reduction approach. Our approach comprises two steps. The first one is the sub-feature selection based on MI, and the second step the SVD approach is applied. The SVD is based on the multi-Eigen value selection which is a new method for dimensionality reduction in data mining and machine learning.The principle idea is that we want to find a non-biased threshold for the subfeature selection. For this purpose, We developed a mathematical model that determines a significant threshold for feature selection based on the MI scoring. For this, We determine the uncertainty of MI scoring-based feature selection. Therefor we select the positive scoring produced by the MI. Afterward, We normalize the whole feature space. To ensure the non-negativity selection for the MI. We assume that the MI measures the inheritance dependence expressed in the joint distribution x and y which is related to the joint distribution of x and y under the assumption of independence. In this case, MI measures the dependency $I(x, y) = 0$ if x and y is independent random variables. That means in the case x and y are independent, the probability of x is represent by p [Pea01] [ZB16] as in equation 1.

$$(x, y) = p(x) \times p(y) \quad (1)$$

Applying the log to the probability results in:

$$log\left(\frac{p(x)}{P(x)P(y)}\right) = log(I) = 0 \quad (2)$$

ISBN 978-80-86943-42-8

That means the MI scoring has non-negative values.

$$I(x,y) \geq 0 \tag{3}$$

Moreover, the MI scoring is asymmetric as shown in Equation 4:

$$I(x,y) = I(y,x) \tag{4}$$

After we ensure that MI is a non-negative value, we need to find the relation between the conditional and the joint entropy. MI can equivalently express by:

$$\begin{aligned} I(x,y) = H(x) - H(y) &\approx H(x) - H\left(\frac{y}{x}\right) \\ &\approx H(x) + H(y) - H(x,y) \\ &\approx H(x,y) - H\left(\frac{x}{y}\right) \\ &\quad - H\left(\frac{y}{x}\right) \end{aligned} \tag{5}$$

where $H(x)$ and $H(y)$ are original entropies, and $H\left(\frac{x}{y}\right)$, $H\left(\frac{y}{x}\right)$ are conditional entropies.

Because of $I(x,y)$ is a non-negative value,

$H(x) \geq H(x/y)$, because of

$$I(x;y) = H(y) - H\left(\frac{y}{x}\right). \tag{6}$$

The proof of this is illustrated in the following:

$$\begin{aligned} I(x,y) &= \sum p(x,y)\, log\left[\frac{p(x,y)}{p(x)p(y)}\right] \\ &= \sum p(x,y)\, log\left(\frac{p(x,y)}{p(x)}\right) \\ &\quad - \sum p(x,y) log p(y) \\ &= \sum p(x) p\left(\frac{y}{x}\right) \log p\left(\frac{y}{x}\right) \\ &\quad - \sum p(x,y) \log p(y) \\ &= \sum p(x) \left(\sum p\left(\frac{y}{x}\right) \log p\left(\frac{y}{x}\right)\right) \\ &\quad - \sum \log p(y) \left(\sum p(y,x)\right) \\ &= -\sum p(x) H\left(\frac{Y}{X=x}\right) - \sum \log p(y)\, p(x) \\ &= -H\left(\frac{Y}{X}\right) + H(y) = H - H\left(\frac{Y}{X}\right) \end{aligned} \tag{7}$$

If the entropy $H(y)$ measures the uncertainty of the random variable, $H\left(\frac{Y}{X}\right)$ measures what X dos not say about Y. This means that the amount of reaming uncertainty of Y after X is known. Then, the amount of MI as the amount of information (that reduction in uncertainty) knowing either variables provide information about the other.

Our Approach for the Singular Value Description (SVD)

Our dimensionality reduction/feature selection approach depends on the regular SVD approach. Our method is based on ranking the eigenvalues first, followed by an accelerated singular value decomposition. For this purpose, multiple eigenvalues have been selected based on ranking step [Pea01]. Our function produces a diagonal matrix S of the dimension as the rank of X (whole data dimensions) with non-negative diagonal elements in decreasing order, and unitary matrices U and V with U represents the eigenvalues, and V represents the eigen vectors.

Algorithm: Our Singular Value Decomposition (SVD)

Input: Generate data matrix $\boldsymbol{X}$
Output: New dimensions $\boldsymbol{C}$
Determine the max matrix size
Determine the eigenvector ratio = 0.1
Compute the $\boldsymbol{S}$ matrix (compute data $\boldsymbol{X^t}.\boldsymbol{X}$)
  Check the optimally reduced dimension
    Compute the diagonal data matrix
    Find the max value of data $\boldsymbol{X}$
    Find the data size
    Check the reduced dimension
      Compute the diagonal eigenvalues
      Sort the value of the eigenvalue
      Get the index of the eigenvalue
      Get $\boldsymbol{U}$ eigenvalue matrix
      Select the max eigenvalue
    Get the index of the eigenvalue
    Replace the eigenvalue with the index
  Select the Eigenvalue
  Compute the unitary matrices $\boldsymbol{U}$
  Produce diagonal matrices of the dimension as the rank $\boldsymbol{X}$ and with non-negative diagonal elements in decreasing order $\boldsymbol{S}$
Find minimum half of the eigenvalue
Compute the unitary matrices $\boldsymbol{V}$ eigenvector

Algorithm 3: Our approach to compute the SVD

4. EXPERIMENTAL RESULTS

In this section, the obtained results of the proposed visual dynamic network evaluated. Our methods are implemented by using Matlab 2017a. For analyzing the proposed dimension reduction, topological observations, mainly a Facebook dataset used. Facebook is one of the most popular social networks that is widely used by millions of people. The used Facebook dataset is given from SNAP consists of 'circles' (or 'friend's lists') from Facebook. The dataset holds node features (profiles), circles, and ego networks. In our experiments, we evaluate our system according to different criteria such as visualizing the whole social network once with each social sub-network as well as the visualization of time series within social networks. Finally, we determine the time-consuming for computing closeness centrality in two cases. First, by using our dimension reduction approach on social network visualization secondly, without dimension reduction.

Whole Network Combination and Visualization

In this section, we visualize the whole social network by one single plot, see Figure.5 the whole network has many sub-networks (ten sub-networks).

 ISBN 978-80-86943-42-8

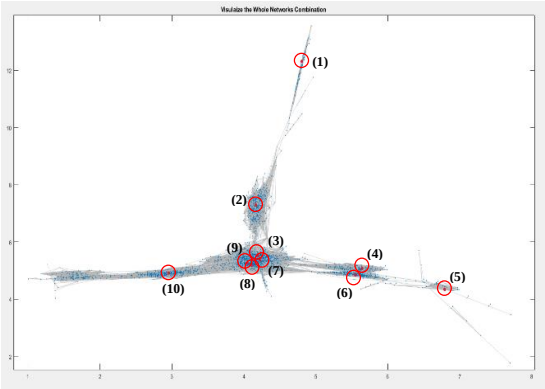


Figure5: Whole social network visualization

In this case, we combine all ten sub-social networks or (ego networks) into one visualization of the social network. The first plot shows that there are more than 4,000 nodes and more than 84,000 edges.

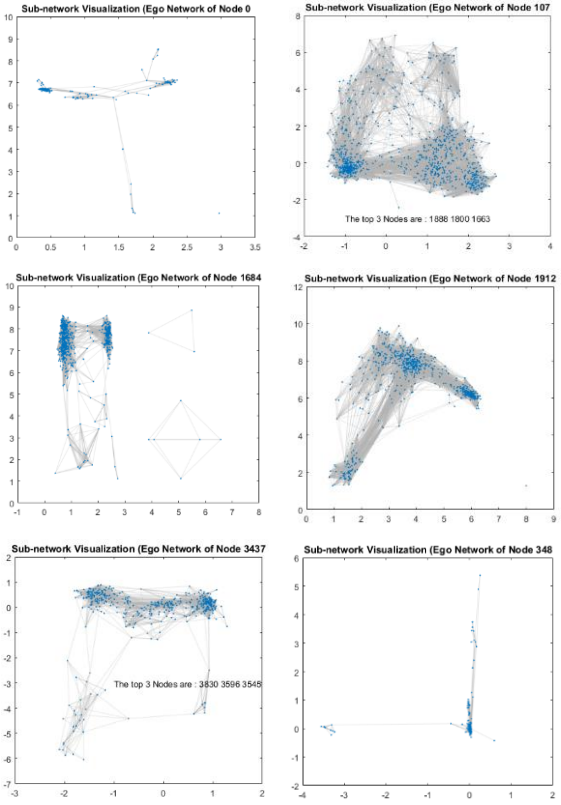


Figure 6: Individual sub-network

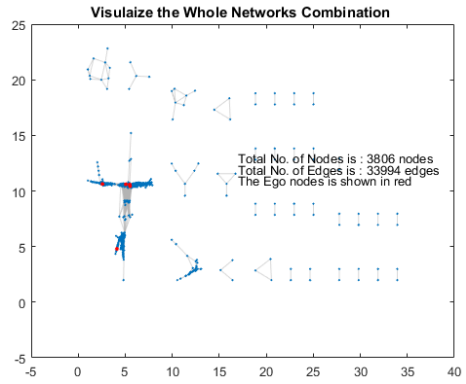
The ten sub-social networks or ego-node that has been marked by red dots

Secondly, every sub-social network (ego-nodes) has visualized individually. Figure.6 shows examples of six sub-networks out of ten. Each sub-social network is visualized.

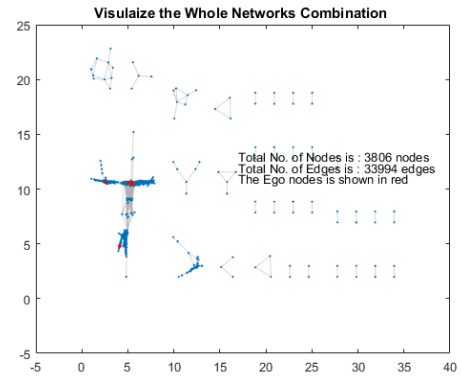
Time Series Social Network Visualization

One of Our main contributions in this paper are the an approach to visualize time series. Our aim to visualize effeteness of the time series overall social network configurations based on the number of connections and the total number of nodes at that time. For this purpose, we pre-processed the Facebook dataset by adding the acquisition time to the main CVS file. That means we collected four updates on the same network according to five-time periods 2008, 2009, 2010, 2011, and 2012. The outcome of our visualization system is a visualization of time series for social networks.

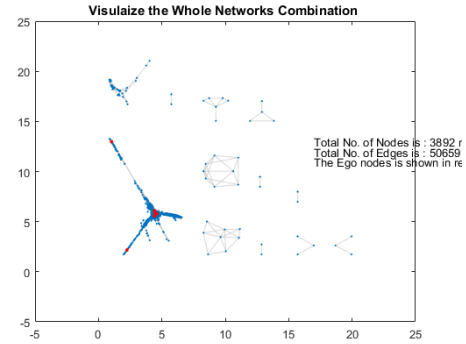
In this case, we performed a variety of Experiments to visualize every single time and the effeteness of this selection on the whole network. Figure.7 shows the effects of the time-series according to the total number of nodes in each selection as well as the total number of edges (connection) in each time selection.



One-year Visualization 2008



Two-years Visualization 2008-2010



Three -years Visualization 2008-2009-2010

ISBN 978-80-86943-42-8

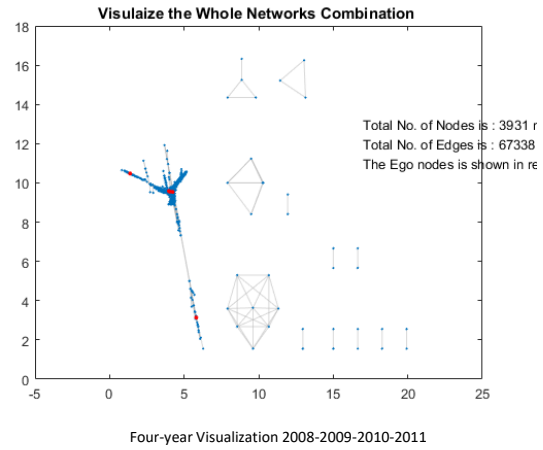


Figure7: Time-series network visualization

Time-series network evolution

Figure.8 shows the main computation (total number of nodes and number of connections) that have changed in the meantime.

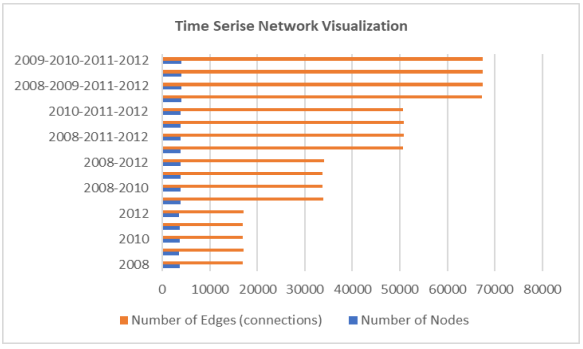


Figure8. A statistical measurement of time-series visualization

It has been shown here that the absolute visualization majority for each time-series that we selected in every single experiment. We can notice that the total number of nodes, as well as the total number of connections, are increased especially when more than one period was selected. For instance, the number of nodes in 2008 was 3583. Then it has been increased to 3940 when we select time series 2008-2012. Also, the total number of connections increased as from 16916 in 2008, to 67400 in the same period (2008 to 2012). This show that the number of connections and nodes is monotonically increased over the five years until 2012.

Degree centrality visualize for single sub-network

As a basic visualization evaluation technique based on the link analysis, this kind of visualization for each sub-network is the most basic analysis that can perform on a social network. With this experiential evaluation, we want to figure out how are the best connected to each network. For this purpose, we change the color of the connection based on the connection number in that sub-network which also represents the connection degree. However, for each sub-network, there is a metric of the connection degree that is known as a connection degree centrality. Figure.9 shows some example of six sub-networks out of ten that depict the centrality degree of each sub-social network.

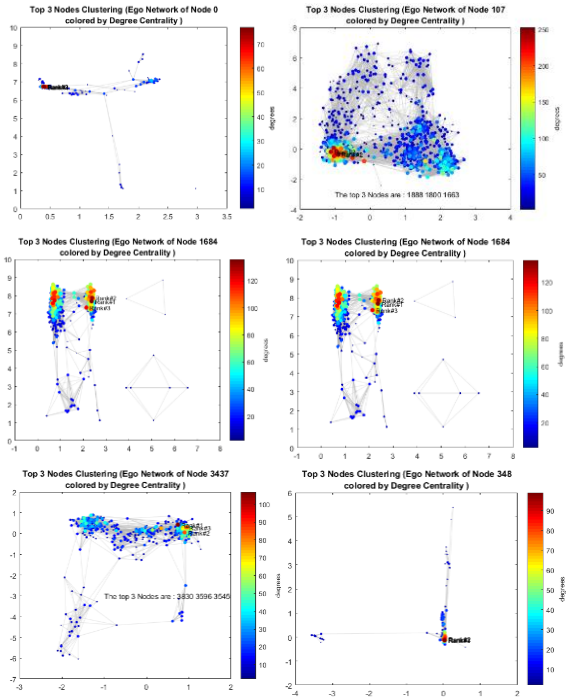


Figure9: Centrality degree visualization for each sub-network

The top three nodes that have been marked by the degree of connection centrality highlighted in the same plot. The nodes are closely connected to each other in the visualized network.

Sub-Social Network Degree Distributed Visualization

Another outcome of our visualization system, we visualize the degrees distribution between each sub-network and whole combined network. For this purpose, the histogram of the degree for each sub-network has been computed and compared with the original network (combined one). People who are active on the social network Facebook have a stronger edge distribution than others. Moreover, a few people have a large number of degrees in a social network such as a Facebook that we want to visualize in our exponential results. The majority, in this case, exhibits small number of degrees and a large difference in the edges distribution. This gives us an indication that is exponential. Figure.10 shows the individual sub-network no. (2) according to the degree centrality visualization and Figure.11 shows the histogram visualization of the degree distribution of the same sub-network no. (2).

ISBN 978-80-86943-42-8

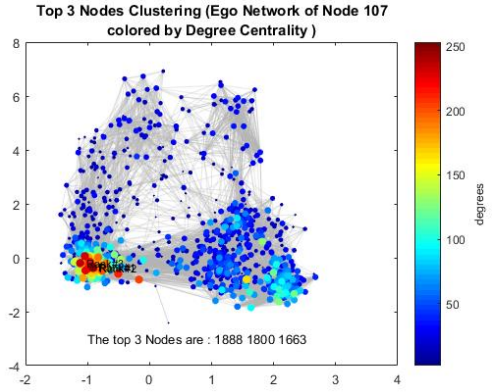


Figure10: Centrality degree visualization of sub-network no. (2)

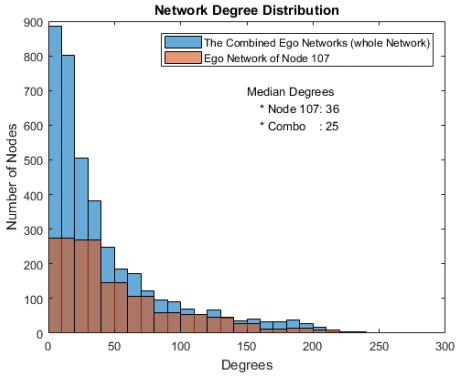
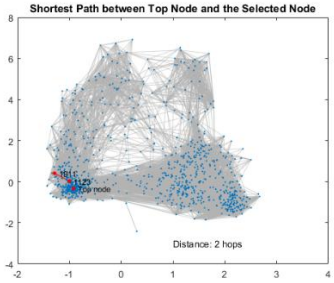


Figure11: Newtown degree distribution of sub-network no. (2)

We can notice that the median degree in the sub-network is less than the degree in the whole combine network. That is because the degree of the node connection in the subnetwork has been only computed based on the node in this individual sub-network. However, in this case, we do not visualize the other nodes that not connected to the other sub-network.

Shortest Path Visualization

The other visualization metric is the degree of the metric evaluation nodes which means more nodes have a better connection. That means the short path indicates how many hops exist in the metric. Figure.11 shows the short path between the top nodes that has been detected in the subnetwork no. (2) and the selected node number 1911, the selected node number describes how many minimum numbers of the node we need to reach our destination although Figure 12 shows the zoomed plot of the short path between the top node and the node number 1911.



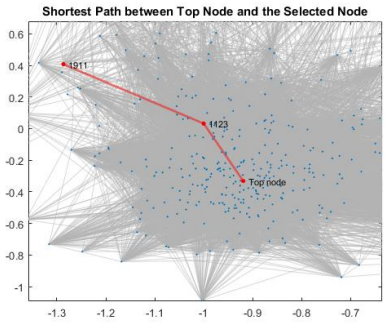


Figure12: Shortest path visualization

Closeness Centrality Visualization

The distances measured by shortest paths been shown in the previous section to be used to compute closeness centrality this is time-consumption issue in the whole network dimension, especially for computing closeness centrality. In this section, we measure the time to compute the closeness centrality for each subnetwork using our dimensionality reduction approach. Table.1.

Sub-Network	Sub-Network Dimensionality	
	Original Feature Space	Reduced Feature Space
1	225	66
2	577	161
3	320	107
4	481	151
5	263	71
6	162	50
7	43	15
8	106	41
9	64	22
10	49	20

Table 1. Dimensionality Reduction for Each Sub-Network

Then, we compare our time with the original dimension for each subnetwork without dimension reduction. Figure.12 shows the results for the time consuming (in minutes) between the approach after dimensionality has reduced for each sub-network and the original feature space. We can notice that the average time consuming for closeness centrality using the original feature space is 52 minutes where our dimension reduction approach consumed 4 minutes. This shows that our proposed approach is significantly faster than the original approach (using the whole feature space for computing closeness centrality).

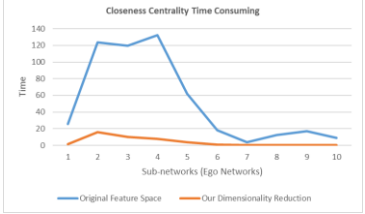


Figure13: Closeness centrality time consumption

ISBN 978-80-86943-42-8

5. CONCLUSION

This paper solves two significant problems of dynamic networks. The problem of dimension reduction and topological observation is resolved using MI and Feature selection based on developing an SVD algorithm. Our results show that the proposed approach for the time-series-network visualization based on the dimensionality reduction is significantly faster than the original dimension for distances measured by shortest paths detection based on the closeness centrality computing

6. ACKNOWLEDGMENTS

This work was partially supported by Ministry of Higher Education and Scientific Research (MHESR), Iraq, University of Koblenz Landau, Germany, HCED (Higher Committee for Education Development in Iraq), and University of Diyala-College of Science, Iraq.

7. REFERENCES

[ARK13] Andry Alamsyah, Budi Rahardjo, and Kuspriyanto. Social Network Analysis Taxonomy Based on Graph Representation. The 5th Indonesian International Conference on Innovation, Entrepreneurship, and Small Business (IICIES), (June):341–349, 2013.

[BJ15] B Birregah and O Jaafar. TVG-OSN: A Time-Varying Graph Model for Online Social Network Dynamics Analysis. Proceedings - 2nd European Network Intelligence Conference, ENIC 2015, pages 17–24, 2015.

[CZ17] Hamideh Sadat Cheraghchi and Ali Zakerolhosseini. Toward a novel art inspired incremental community mining algorithm in the dynamic social network. Applied Intelligence, 46(2):409– 426, 2017.

[FPA+12] Paolo Federico, Ju¨rgen Pfeffer, Wolfgang Aigner, Silvia Miksch, and Lukas Zenk. Visual Analysis of Dynamic Networks Using Change Centrality. Asonam, pages 179–183, 2012.

[FS15] Norie Fu and Vorapong Suppakitpaisarn. Clustering 1-Dimensional periodic network using betweenness centrality. Lecture Notes in Computer Science, 9197:128–139, 2015.

[GSZ+11] Frederic Gilbert, Paolo Simonetto, Faraz Zaidi, Fabien Jourdan, and Romain Bourqui. Communities and hierarchical structures in dynamic social networks: analysis and visualization. Social Network Analysis and Mining, 1(2):83– 95, 2011.

[LCZ+17] Xuefei Li, Lijun Chang, Kai Zheng, Zi Huang, and Xiaofang Zhou. Ranking weighted clustering coefficient in large dynamic graphs. World Wide Web, 20(5):855–883, 2017.

[Pea01] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6, 2(11):559–572, 1901.

[SVM14] C O S Sorzano, J Vargas, and a Pascual Montano. A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877, pages 1–35, 2014.

[WCG+16] Yingcai Wu, Nan Cao, David Gotz, YapPeng Tan, and Daniel A. Keim. A Survey on Visual Analytics of Social Media Data. IEEE Transactions on Multimedia, 18(11):2135–2148, 2016.

[Wu10] Yu Wu. Pattern Analysis in Dynamic Social Networks. 117(May):43–48, 2010.

[XH16] Wang Xiangang and Song Hanchen. A Dynamic Network Layout Visualization Method Based on Structural Similarity. 2016 International Conference on Cyberworlds (CW), pages 119–126, 2016.

[ZB16] Charles Y. Zheng and Yuval Benjamini. Estimating mutual information in high dimensions via classification error. (Nips), 2016.

 ISBN 978-80-86943-42-8

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

Deep Learning for Historical Cadastral Maps Digitization: Overview, Challenges and Potential

Jelena Ignjatić
Urban and Spatial Planning Institute of Vojvodina
Zeleznicka 6/II
21000, Novi Sad, Serbia
ignjaticjelena@gmail.com

Bojana Nikolić
University of Novi Sad,
Faculty of Technical Sciences
Trg Dositeja Obradovica 6
21000, Novi Sad, Serbia
bojana.nikolic.ftn@gmail.com

Aleksandar Rikalović
University of Novi Sad,
Faculty of Technical Sciences
Trg Dositeja Obradovica 6
21000, Novi Sad, Serbia
a.rikalovic@uns.ac.rs

Dubravko Ćulibrk
University of Novi Sad,
Faculty of Technical Sciences
Trg Dositeja Obradovica 6
21000, Novi Sad, Serbia
dculibrk@uns.ac.rs

ABSTRACT

Cartographic heritage of historical cadastral maps represent remarkable geospatial data. Historical cadastral maps are generally regarded as an essential part of the land management infrastructure (buildings, streets, canals, bridges, etc.). Today these cadastral maps are still in use in a digital raster form (scanned maps). Digitization of cadastral maps is time consuming and it is a challenge for scientists and engineers to find ways to automatically convert raster into vector maps. The process of map digitization typically involves several stages: preprocessing, visual object detection and classification, vector representation postprocessing and extracting information from text. Although neural networks have had a long history of use in the domain, their applications remain limited to extracting the information from text. Recent convergence of advancements in the domains of training deep neural networks (DNN) and GPU hardware allowed DNNs to achieve state-of-the-art results in computer vision applications, beyond hand-written text recognition. This paper provides an overview of different approaches to historical cadastral maps digitization, focusing of the challenges and the potential of using deep neural networks in map digitization.

Keywords

Convolutional Neural Networks, Deep Neural Networks, Map Digitization, Map Vectorization, Pattern recognition

1. INTRODUCTION

Cadastral plans represent a comprehensive register of parcels and facilities of a certain area with related information [Vas16]. Cartographic heritage of historical cadastral maps of the Habsburg Empire (i.e. Austro-Hungarian Monarchy) from XVII and XIX century represent remarkable geospatial data in Europe. The territory of Serbia, which was a part of the Austro-Hungarian Monarchy prior to the First World War, had the geodetic plans made by the graphic method, and cadastral plans were done based on them. In many parts of the territory of Serbia, Bosnia and Herzegovina and Montenegro old Austro-Hungarian cadastral plans are still in use, divided into sheets according to the regulations that were applied, at that time, in Austria-Hungary [Sap17]. The cadastral maps for parts of Serbia's Northern Province of Vojvodina consist of such analogue maps, which were scanned to obtain raster images and then georeferenced.

While these images are already in digital form, the full process of digitization usually represents the process of conversion of an analogue map into a vector graphics map with all of its associate data [Vas16]. The shift from raster to vector representation is the more challenging part of the process. Therefore, in the study presented here, we focus mainly on this part of the process of map digitization, where the input to the process are digital images representing cadastral maps from XIX century.

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.6

Cadastral maps in vector form have many advantages over the raster ones: faster data processing is achieved, map modification and replenishment is easier, and less storage space is required.

The problem of complete automatization of the conversion of raster maps into vector form has not been adequately solved so far and continues to attract research interest. The literature on the methods for automatic conversion of cadastral plans from raster to vector form has a long history. In the past 30 years, several automatic systems have been developed for digital map processing. Works related to the automatic vectorization of cadastral maps date back to the 1990s [Eji90], [Jan93], [Che96], [Kat99].

Although neural networks have been used in this domain for a long time [Che96], [Kar08], they have recently experienced an expansion in the form of deep learning, achieving significant improvements over the state-of-the-art in a number of applications. Deep Neural Networks (DNNs) have demonstrated prepossessing and even human-competitive results on many object and pattern recognition tasks, especially when it comes to vision classification problems. However, a review of the relevant literature reveals that the applications of deep learning methods to problem of the vectorization of cadastral plans have been very limited [Oli17].

Map digitization is a procedure which requires visual object and pattern recognition, such as the identification of edges (parcel and building lines), symbols, text, and patterns – areas in which DNNs achieve excellent performance.

In the study presented here, we will strive to evaluate the potential benefits that can be achieved through a more extensive use and application of deep learning technology to the problem of vectorization of cadastral maps automatically, or with minimal human intervention. As an aid for researchers the paper presents an overview of the problem, existing state-of-the-art solutions proposed and, more importantly, presents a detailed discussion of state-of-the-art deep learning architectures and approaches that can readily be applied to the domain and or are likely to achieve best results.

2. RELATED WORK

In earlier times, geographic data sets were used only in their analogue form. Today, with the development of technology, geographic information is mainly available in the form of digital data sets.

The problem of digitizing geographic data sets is complex. To illustrate the problem, this paper focuses on digitization of cadastral maps done under the reign of Mary Theresa [McG66]. Fig.1 shows a part of the original scanned cadastral map, while Fig.2 corresponds to the same map in vector form.

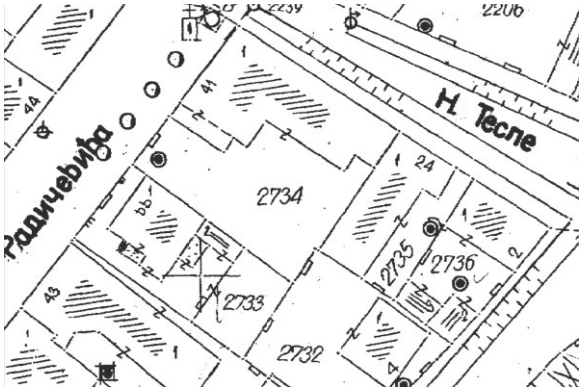


Figure 1. An original scanned cadastral map from the time of Mary Theresa

Figure 2. A sample from a vectorized cadastral map from the time of Mary Theresa (corresponding to the map in Figure 1)

There are many studies that deal with map vectorization [Boa92], [Ill91], [Con97], but the first that attempts to use neural networks to solve the complex problem of the digitization of Chinese cadastral maps and automatic extraction of high-level information was done by Chen et al. in 1996 [Che96]. The system they had developed at the time, consisted of three main parts: the separation of text and graphics, extraction of parcels and recognition of rotated characters. Even though the text and graphic separation and parcels extraction was not done by machine learning methods, they were among the first ones to propose a neural network approach to the recognition of Chinese hand-written and rotated characters. The used neural network was a two-dimensional Hopfield neural network [Hop82] which was able to reflect the degree of similarity of the test characters to previously stored templates in the final states of neurons.

Several years later, Katona and Hudra proposed a slightly different pipeline to deal with the issue of cadastral map vectorization [Kat99]. The authors of the paper developed an application based on the approach, dubbed MAPINT. The main processing phases of their approach are: raster-to-vector conversion, segmentation,

ISBN 978-80-86943-42-8

recognition and automatic correction of vectorization anomalies. The raster-to-vector conversion is the phase in which a raw vector image is generated from the scanned map as a skeleton image, through line thinning and line tracing. In the segmentation phase, raw vectors are classified into three basic categories: (text) symbols, dashed lines and continuous lines. The recognition phase is divided into three steps: recognition of symbols, recognition of connection-signs and recognition of buildings and parcels. Only the recognition of symbols was done by a neural network. The full architecture of the net is not specified by the authors, other than that it is trained using back-propagation.

Like Chen et al. [Che96], Katona and Hudra [Kat99] were using the neural network for recognition of rotated hand-written symbols which were pre-segmented as a groups of symbols, i.e. strings. The contribution of their paper is reflected in the distinction between house numbers and parcel numbers, where the distinction is made by size and length of number strings.

In 2008, Karabork et al. [Kar08] used a mulit-layer perceptron neural net to trace lines and detect joints in a skeleton image obtained from scanned maps. The study is remarkable in tha it seems an only relatively early attempt to use neural nets for more than text recognition in the domain of map digitization, but their approach failed to improve on Sparse Pixel Vectorization which was proposed some years earlier [Wen97].

Recently, Oliveira et al. [Oli17] discussed the implementation of a fully automated process based on machine learning algorithms for Napoleonic cadastral map digitalization. They proposed a system for the vectorization of the cadastral map, where their methodology consists of several steps: preprocessing, segmentation, regional classification, parcel extraction, digit extraction and digit recognition.

Once basic preprocessing is done, in order to eliminate variations in hue present in scanned maps, regions are extracted in the form of merged super pixels and a Support Vector Machines (SVM) classifier is used to classify the regions into 3 classes: text, contour and background.

In the final step of their approach, a deep Convolutional Neural Network (CNN), called LeNet [LeC98], is used to recognize digits. The network was used "out-of-the-box", without any additional training or fine tuning, was pre trained on the MNIST dataset [LeC98], and was used only to recognize digits in their pipeline.

This may have contributed to the fact that their method achieved an accuracy of only 10% for identifying hand-written numbers, although the authors propose that this was not due to the recognition algorithm itself, but rather because of an insufficiently good digit segmentation procedure.

3. DEEP LEARNING FOR MAP DIGITIZATION

With the advent of the age of Big Data and the development of new technologies, in recent years deep learning methods have been shown to outperform previous traditional state-of-the-art machine learning techniques in several fields, including computer vision [Vou18].

As we are moving towards more complete computer vision, full historic map understanding becomes achievable, as the crucial steps in this process are visual object and pattern recognition, which is becoming more precise and detailed. Recent years have produced great advances when it comes to training large deep neural networks. However, deep architectures, which are characterized by many building blocks [Yos15], [Pra18], [Cir12], require significant computational resources for training. A fact that was a limiting factor for the advancement of deep learning over several years.

The problem has been remedied by significant progress made in the development of Graphics Processing Units (GPU), which accelerate the whole process and enable the training of large DNN to last for days, instead of months [Cir12]. Today, DNNs represent the state-of-the-art in object and pattern recognition and one can expect the full integration of deep learning technology in the domain of map processing in the near future.

In comparison with more "classical" (shallow) Neural Network (NN) architectures, DNNs are able to learn high-level features, which are more complex and abstract, by integrating feature learning and model construction into a single model. That model is created by selecting different kernels or tuning the parameters by end-to-end optimization with the minimum human inference. where the parameters of DNN model are trained jointly [He15], [Sze17]. Its deep architecture, with many hidden layers, is a multi-level non-linear operation, transferring each layer's features from original input into more abstracted features in the higher layers to find complicated inherent structures. Unlike DNNs, shallow NNs approaches the extraction of the features and the construction of the model are done separately, where each module is formed by step-by-step model training [Wan18].

In the rest of this section, we provide a description of CNN, which is the common DNN method used for computer vision and image recognition.

 ISBN 978-80-86943-42-8

Map Digitization: CNN approach

To learn about objects and patterns from a cadastral maps, a model with a large learning capacity is needed. The last several years have produced tremendous progress in training powerful DNN models, which are used for challenging computer vision problems. The applications of DNNs to computer vision tasks usually rely on a special form of DNNs, called Convolutional Neural Networks (CNN) [Mur10], which were firstly proposed by LeCun and Bengio [LeC95].

A CNN is a multi-layer feed-forward artificial deep neural network that has fewer connections and parameters than other NN architectures originally intended for two-dimensional image processing [LeC98], [Wan18], [Kri12].

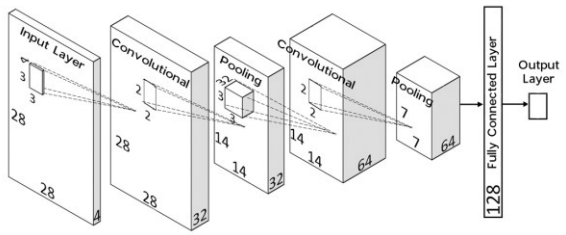


Figure 3. Structure of CNN (adopted from [Che11])

The structure of CNN is illustrated in Figure 3. It consists of three main hidden layer types: convolutional layers, pooling layers and fully connected layers [Vou18]. According to Voulodimos et al. [Vou18] and Cireşan et al. [Cir11], the convolutional layers are parameterized by the size and the number of the feature maps, kernel sizes, skipping factors and the connection table.

In CNN, the pooling layers enable position invariance over larger local regions and summarize the outputs of neighboring groups of neurons in the same kernel map, where the neighborhoods, summarized by adjacent pooling units, do not overlap [Kri12], [Cir11]. The fully connected layers are used for high-level reasoning in the neural network [Vou18] which combine the outputs of the top convolutional layer and convert the 2D feature maps into a 1D feature [Vou18], [Cir11].

The way of achieving the feature learning in CNN is by alternating and stacking convolutional layers and pooling operations. The convolutional layers convolve multiple local kernel filters with input data in order to generate invariant local features [Wan18]. The pooling layers extract the most significant features over sliding windows, which have a fixed-length using different pooling operations. Typical pooling operations are average pooling and max pooling. Average pooling calculates the mean value of the neurons in the region (sliding window) and takes it as the output value for that region [Wan18]. Not surprisingly, max pooling selects the maximum value in the region, so the pooling output is the maximum activation over rectangular regions that do not overlap [Cir12].

Max pooling is well suited to extract sparse features, while other operations might not be optimal on all samples [Wan18]. In comparison with similar models, CNN has several advantages, including easier training, sparse interactions with local connectivity, reduced number of parameter sharing and equivariant representation which is invariant to object locations [Wan18], [Kri12].

4. CHALLENGES AND POTENTIAL OF DNN IN MAP DIGITIZATION

In this section, papers that are surveyed have leveraged deep learning methods to address the main tasks in computer vision, in order to digitized maps, such as object and pattern recognition.

A survey of the literature of done in this study lead to two main findings.

The first is the fact that the different approaches to the problem of extracting vector graphics representations of historical cadastral plans and maps follow a relatively general pipeline, which can be broadly defined to contain the following stages: preprocessing, visual object detection and classification, vector representation postprocessing and extracting information from text. The last two stages are independent and can be done in parallel.

The second finding is that, although neural nets have had a long history of use in these approaches, their applications have mainly been limited to the stage of extracting the information from text and that this has remained so, even with the advancement of deep learning. This, perhaps, is not surprising when one considers that the convergence of advancements in the domain of training deep neural nets and GPU hardware has only recently (as of 2012 [Kri12]) allowed DNNs to achieve state of the art results in computer vision applications, beyond hand-written text recognition.

This state of affairs is likely to end soon. A recent (2017) study by Liu et al. [Liu17] addresses the problem of converting a rasterized two-dimensional image into a vectorized representation using a CNN. Evenhough the floor plan images are used, instead of the maps, their approach is perhaps the one that can most readily be transferred to the domain of vectorizing maps.

The approach of Liu et al. consists of two computational machineries (deep representation learning and integer programming) combined into a

 ISBN 978-80-86943-42-8

single system that converts a floor plan image through two intermediate representation layers. First, a CNN converts a floor plan image into a junction layer, where data is represented by a set of junctions identified as specific objects by the network. Secondly, integer programming aggregates junctions into a set of primitives, while providing a topologically and geometrically consistent result. After that, a simple post-processing is used for producing the final vector format.

In terms of the general pipeline for cadastral data vectorization, the deep representation learning stage corresponds to the visual object detection and classification and focuses on detecting key points (line joints, types, etc.), while integer programming deals with vector representation postprocessing and is used to construct a topologically and geometrically consistent result. Of the two remaining tasks necessary to achieve fully automated cadastral plan digitization, extraction of the information from hand-written text has always been the domain of (deep) neural networks. Finally, the preprocessing was mainly done to simplify the task of visual object detection and eliminated "noise" present in old maps (such as accidental blotches and variations in color). While some preprocessing might still be meaningful when a full DNN framework for the vectorizatoin is considered, the success of DNNs in the domain of computer vision was in a significan part due to their invariance to such distortions in the input data and it is likely that preprocessign will be eliminated from the pipeline all together.

The main hurdle in terms of using DNNs for historic cadastral plan vectorization is likely to be a limited amount of training data available. E. g. the study of Liu et al. [Liu17] used the 100 870 data set, based on LIFULL HOME'S dataset [Lif18], containing 870 human-annotated floor-plan images and 100,000 unannotated test images to train their DNN. Generating such a dataset for historical cadastral data seems not only prohibitively expensive, but is also likely to be unjustified, when one considers the potential applications of the technology. To make matters worse, the results of Oliviera et al. [Oli17] suggest that using existing pre trained models can lead to a significant degradation of performance (10% vs. over 99% achieved on the MNIST data set the network was trained and tested on [LeC98]).

At the time of writing, the Generative Adversarial Networks approach [Goo14], [Cre16] seems to be the state-of-the-art alternative that could help achieve good performance, even in the absence of labelled training data. In addition, the floor plan dataset by Liu could be used to pre trained the models, and fine-tune them on the cadastral data, as the two domains are different but relatively similar.

Although some challenges remain, a wider adoption of deep learning technology in the process of map and cadastral data vectorization seems immanent and is likely to revolutionize the field in the near future, as it has already done for so many other domains.

5. CONCLUSION AND FUTURE WORK

Map digitization represents a procedure which relies on visual object and pattern recognition, such as the identification of parcel and building lines, line junctions, symbols and text – areas in which DNNs achieve excellent performance. However, over the past three decades, not many studies dealing with the problem of map vectorization problem used NNs outside the scope of written-text recognition. Deep learning, in particular, has barely entered the domain and the survey of literature conducted for the purposes of the study presented here identified only one approach that used a CNN to extract information from text, while relying on more classical computer vision for the rest of the process.

Recognizing that new advances in the domain of deep learning, which have already revolutionizes many areas of computer vision, are likely to be applied to the problem of cadastral map vectorization in the near future, we have attempted to assess the challenges in this process and the potential benefits that can be achieved, as an aid for researchers considering this course of action and in order to stimulate further discussion and a possible faster adoption of the technology. Having identified the challenges, we've also attempted to identify specific approaches in the domain of deep learning that can be used to address them.

Our findings presented in the paper result from the initial stage of the study aimed at achieving beyond-state-of-the-art automatic historical cadastral plan vectorization through the use of deep learning.

REFERENCES

[Boa92] Boatto, L. et al. An interpretation system for cadastral maps. Computer, vol. 25, no. 7, pp. 25–33, 1992.

[Che11] Chen, B. X., Sahdev, R. and Tsotsos, J. K. Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot. Internacional Coference on Computer Vision Systems, pp. 300–313, 2011.

[Che96] Chen, L.-H., Liao, H.-Y., Wang, J.-Y., Fan, K.-C. and Hsieh, C.-C. An interpretation system for cadastral maps. Proceedings of 13th International Conference on Pattern Recognition, pp. 711–715, 1996.

[Cir11] Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M. and Schmidhuber, J. Flexible, high performance convolutional neural networks

 ISBN 978-80-86943-42-8

for image classification. IJCAI International Joint Conference on Artificial Intelligence, pp. 1237–1242, 2011.

[Cir12] Cireşan, D., Meier, U., Masci, J., and Schmidhuber, J. Multi-column deep neural network for traffic sign classification. Neural Networks, vol. 32, pp. 333–338, 2012.

[Con97] Congalton, R. G. Exploring and Evaluating the Consequences of Vector-to-Raster and Raster-to-Vector Conversion. Photogramm. Eng. Remote Sens., vol. 63, no. April, pp. 425–434, 1997.

[Cre16] Creswell, A. and Bharath, A. A. Adversarial training for sketch retrieval. Computer Vision – ECCV 2016 Workshops, vol. 9913 LNCS, pp. 798–809, 2016.

[Eji90] Ejiri, M., Kakumoto, S., Miyatake, T., Shimada, S. and Iwa-mura, K. Image analysis application, 1st ed. 1990.

[Goo14] Goodfellow, I. et al. Generative Adversarial Nets. Advances in Neural Information Processing Systems 27, pp. 2672–2680, 2014.

[He15] He, K. and Sun, J. Convolutional neural networks at constrained time cost. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5353–5360, 2015.

[Hop82] Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, vol. 79, no. 8, pp. 2554–2558, 1982.

[Ill91] Illert, A. Automatic Digitization of Large Scale Maps. Building, pp. 113–122, 1991.

[Jan93] Janssen, R. D. T., Duin, R. P. W. and Vossepoel, A. M. Evaluation method for an automatic map interpretation system for cadastral maps. Proc. 2nd Int. Conf. Doc. Anal. Recognit. (ICDAR '93), no. November, pp. 125–128, 1993.

[Kar08] Karabork, H., Kocer, B., Bildirici, I. O., Yildiz, F. and Aktas, E. A neural network algorithm for vectorization of 2D maps. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XXXVII, no. B2, pp. 473–480, 2008.

[Kat99] Katona, E. and Hudra, G. An interpretation system for cadastral maps. Image Analysis and Processing, 1999. Proceedings. International Conference on Image Analysis and Processing, 1999.

[Kri12] Krizhevsky, A., Sutskever, I. and Hinton, G, ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS2012), pp. 1–9, 2012.

[LeC95] LeCun, Y. and Bengio, Y. Convolution Networks for Images, Speech, and Time-Series. Handb. brain theory neural networks, no. 1, pp. 1–5, 1995.

[LeC98] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based Learning Applied to Document Recognition. Procedings of the IEEE, pp. 2278–2324, 1998.

[Lif18] Lifull Home's dataset." [Online]. Available: http://www.nii.ac.jp/%0Adsc/idr/next/homes.htm %0A [Accessed: 18.3.2018.]

[Liu17] Liu, C., Wu, J., Kohli, P. and Furukawa, Y. Raster-to-Vector: Revisiting Floorplan Transformation. 2017 IEEE Int. Conf. Comput. Vis., pp. 2214–2222, 2017.

[McG66] McGuigan, D. G. The Habsburgs, First Edit. Doubleday, 1966.

[Mur10] Murray, J. F. Convolutional Networks Can Learn to Generate Affinity Graphs for Image Segmentation. Neural Comput., vol. 22, no. 2, pp. 511–538, 2010.

[Oli17] Oliveira, S. A., Kaplan, F. and Di Lenardo, I. Machine Vision on Cadaster Plans. Premiere Annual Conference of the International Alliance of Digital Humanities Organizations 2017, 2017.

[Pra18] Prakash, C. Ranking with Deep Neural Networks. Fifth International Conference on Emerging Applications of Information, no. January, 2018.

[Sap17] Sapkota, R. K. and Bha, G. P. Technical Aspects of Digitization of Cadastral Maps. Nepal. J. Geoinformatics, vol. 13, pp. 42–50, 2017.

[Sze17] Sze, V., Chen, Y.-H., Yang, T.-J. and Emer, J. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Proceedings of the IEEE, pp. 1–32, 2017.

[Vas16] Vasiljević, S., Amović, M. and Višnjić, R. Katastarski planovi prema podijeli bečkog vojnogeografskog instituta. Vještak, vol. 2, no. 2, pp. 219–223, 2016.

[Vou18] Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E. Deep Learning for Computer Vision : A Brief Review. Comput. Intell. Neurosci., vol. 2018, 2018.

[Wan18] Wang, J., Ma, Y., Zhang, L., Gao, R. X. and Wu, D. Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems, no. January, 2018.

[Wen97] Wenyin, L. and Dori, D. A protocol for performance evaluation of line detection algorithms. Mach. Vis. Appl., vol. 9, no. 5–6, pp. 240–250, 1997.

[Yos15] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H. Understanding Neural Networks Through Deep Visualization. Deep Learning Workshop, 31st International Conference on Machine Learning, 2015.

 ISBN 978-80-86943-42-8

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

A Persistent Naming System Based on Graph Transformation Rules

Marcheix David
LIAS, ENSMA
France, 86360,
Futuroscope
david.marcheix@
ensma.fr

Cardot Anaïs
XLIM Lab.
France, 86360,
Futuroscope
anais.cardot@
univ-poitiers.fr

Skapin Xavier
XLIM Lab.
France, 86360,
Futuroscope
skapin@xlim.fr

Dieudonné-Glad Nadine
HeRMA Lab.
France, 86000, Poitiers
nadine.dieudonne.glad@
univ-poitiers.fr

ABSTRACT

3D modeling for Archaeology requires to easily model scenes by letting users evaluate a parametric specification of archaeology-oriented gestures, then modify and reevaluate the specification to produce various restitution hypotheses. But the current modeling tools that support reevaluation mechanisms are not dedicated to Archaeology. The *Jerboa* library, based on graph transformations rules, is well suited for creating operations fitting the needs of archaeologists. But it does not any support reevaluation mechanism and especially the *persistent naming system*, that is used to identify the entities of the initial model and match them with entities of the reevaluated model. In this paper, we extend *Jerboa* with a new application-independent persistent naming model, which is more general and homogeneous than other solutions found in the literature and is the first one to handle parametric specification edition.

Keywords

Parametric Specification; Persistent Naming; Graph Transformation Rules; Generalized Maps.

1 INTRODUCTION

Digital Humanities, and 3D modeling tools in particular, have profoundly modified the discipline of archaeology in several ways. They enrich the patrimonial description and significantly improve its understanding by the public. Modeling ancient buildings in 3D usually borrows from: (1) *Computer Vision*, requiring buildings in good condition for 3D replication and/or completion [GBS14]; (2) *Geometry Modeling* based on fragmentary data, which requires the definition of several restitution hypotheses, and the availability of a tool to test these hypotheses quickly and simply. Our work is set in this latter context.

Procedural generation grammars is a commonly used process for creating several variants of the same building [HMV09], [QB15], but requires some rich information corpus information to produce grammars. Moreover, the same tool cannot be used for very different case studies with many specific features. Therefore, archaeologists usually use more "conventional" 3D tools such as *CityEngine*^{TM}or *Blender*^{TM}. Unfortunately, those tools do not comply with inherently incomplete archaeological data [Wit13], [Ver10]. In particular, they cannot easily model a display of several reconstruction hypotheses, each of them matching the observed data. The central problem of testing reconstruction hypotheses on a 3D view basis leads to limited interpretations of the Past, all the more for protohistory, for which remains are scarce.

To overcome these limitations, we use the *Graph Transformation Rules* formalism [EEPT06] through a *Java* library called *Jerboa* [JER], and designed to assist the development of application-specific modelers. Rule-based languages form a standard approach for geometric modeling, from plant growth with the seminal L-Systems [PH89], to numerous applications such as buildings [HMV09]. Unlike most approaches, Jerboa is independent from any application domain and avoids any hand-coding of operations, except rule writing. It allows rapid development of new operations to automatically check the consistency of different objects properties. All applications developed with Jerboa library share the same topological model called *Generalized maps* (or "*G-Maps*") [Lie91], describing a particular class of labeled graphs.

But Jerboa does not support the rapid production of restitution hypotheses, i.e. the mechanisms of *reevaluation* inherent to parametric systems used in CAD domain. Reevaluation allows to modify any part of an

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.7

object construction history and to replay this history to produce a new result. A *parametric system* is a two-fold data structure composed of a geometric model defining the explicit geometry of the designed object (called *parametric object*), and a mechanism able to reevaluate it when some parameters are changed (called *parametric specification*)[Kri95]. The geometric model is usually a topological-based one. Most current parametric modeling systems are known as "history-based" because the parametric specification may be regarded as a history of modeling functions (or *constructive gestures*), which are attached via their parameters to topological entities defined in previous states of the model. Such an approach requires to define how to ensure the persistence of the referenced entities and to avoid systems failure during the reevaluation phase when various kinds of topological changes occur. This issue, known as *persistent naming*, should enable both unambiguous identification of initial model entities and consistent matching between initial and reevaluated model entities.

Persistent naming is a much-debated problem in CAD domain [Kri95] [Bab10][XJHY16], but has never been investigated in conjunction with graph transformation rules. Our approach enables: (1) to extend the persistent naming scope to modeling systems based on such graph transformation rules; (2) to extend Jerboa by including the working mechanisms of parametric systems. We address naming problems through a very precise characterization of the basic elements forming the model and propose a naming mechanism both general (independent of the model dimension) and homogeneous (independent of the entity dimension), for which only the entities actually used in the parametric specification are followed. Unlike others methods, this follow-up is performed only during reevaluation (and not also during initial evaluation), in order to optimize both time and memory consuming. Moreover, beyond static reevaluation with only parameter modifications, we explore how to carry out *parametric specification edition* (i.e. adding or deleting constructive gestures).

In Section 2, we present the G-maps model, the graph transformation rules and our contribution to persistent naming. In Section 3, we detail the different parts of our works, from the persistent naming system to the complete edition of a parametric specification using bulletin boards and history records. We conclude in Section 4 and propose some perspectives.

2 MAIN CONCEPTS

2.1 Generalized maps

As stated above, Jerboa is based on *G-Maps*, which intuitively represent the decomposition of *n*-dimensional objects according to the successive dimensions of their boundaries, the different parts being linked by relationships noted α_i. For example, the 2D object in Figure 1(a) is split into faces linked by α_2 (blue line, Figure 1(b)); face sides are split into edges linked by α_1 (red lines, Figure 1(c)); and ends of edges are linked by α_0 (black lines, Figure 1(d)). A G-Map is therefore a graph whose nodes are called *darts* (represented as green disks in Figure 1(e)) and arcs represent various α_i. Entities are described as specific set of darts linked by dimension-specific α_i: vertices (dim 0), edges (dim 1) and faces (dim 2) are respectively defined as set of darts linked by [α_1, α_2], [α_0, α_2] and [α_0, α_1].

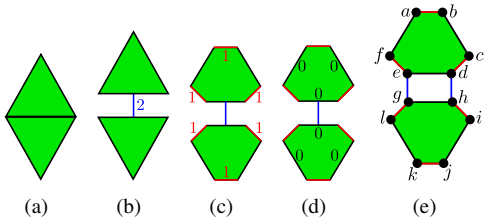


Figure 1: Modeling 2D objects using G-Maps.

We call *orbit type* the set $\{\alpha_i, ..., \alpha_n\}$ describing any entity, denoted as $\langle i...n \rangle$: orbit type "Vertex" (resp. "Edge", "Face") shown in Figure 1 is thus denoted as $\langle 12 \rangle$ (resp. $\langle 02 \rangle$, $\langle 01 \rangle$). We call *orbit* the association of a dart with an orbit type to designate a specific entity. For example, on Figure 1(e), darts {a,b,c,d,e,f} represent a face, {f,e,g,l} a vertex, and {h,i}, the restricted corner of a face. Entities used in our parametric specifications are expressed as orbits. They can be fully characterized by their type and a selection of their darts.

2.2 Graph transformation rules

Jerboa is based on topological rules of graph transformation [BALB14]. Each modeling operation is formally defined as a rule applied to a G-Map. Jerboa ensures by design that the topological consistency of the G-Map is maintained after each rule application.

Rules are made up of two parts separated by a left-to-right arrow. The left (resp. right) part, which describes the pattern to be filtered (resp. the rewritten pattern), represents the model before (resp. after) application. Patterns are defined by the orbit types of the rule nodes. For example, the Vertex Insertion rule is illustrated in Figure 2. The left node n_0 carries the orbit type $\langle 02 \rangle$, and thus filters the edge associated with this node.

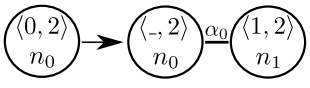


Figure 2: Vertex Insertion rule.

2.3 Persistent naming

Our method of persistent naming is grounded on both G-Maps and rewriting rules. Persistent naming allows

ISBN 978-80-86943-42-8

to characterize the topological entities in a sufficiently robust way during the initial construction. Parameters of parametric specification operations are often topological references, so this mechanism is essential to produce a valid reevaluation.

Various naming methods have been proposed to try and solve this problem in a full and homogeneous way. Most methods ([Kri95][WN05] [XJHY16]) use faces as references to name all other entities, since in 3D, each entity can be characterized by an intersection of faces and some additional geometric information. However, these naming algorithms are *not generalizable in dimension n*. Moreover, the naming mechanism of any entity depends on its dimension, so *the naming is not truly homogeneous*. In addition, even though the design of persistent naming is well depicted in the literature, the way it can be used for reevaluation is not always precisely defined. Furthermore and despite memory overload, it is usually necessary to trace the evolution of many entities during the initial construction, in order to perform the match between entities when reevaluating, even though many of them will not be used.

Finally, based on the review of existing literature, no method explains how to deal with parametric specification *editing*, i.e. adding or deleting gestures between the initial evaluation and the reevaluation. We describe in the next section the various mechanisms that address these limitations.

3 REEVALUATION MECHANISMS

3.1 Parametric specification and edition

To reevaluate a sequence of constructive gestures, we record them in the form of a parametric specification beforehand. Each gesture corresponds to the call of a graph transformation rule as defined in Section 2.2. Let us consider the sequence of gestures performed in the initial specification shown in Figure 3. The specification cannot be limited to the simple recording of rule calls (physical id. of darts being inherently unstable from one reevaluation to another, they cannot be used directly). Darts should therefore be labeled persistently. The use of rules makes it possible, both in the initial evaluation and the reevaluation, to assign each dart a *Persistent Id*, denoted as PI_a, PI_b and so on.

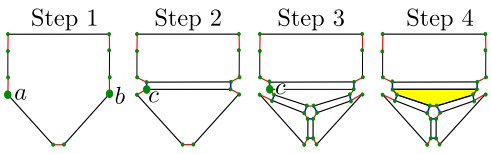


Figure 3: Initial specification.

Rules are defined for any filtered orbits, but only specific orbits are used by gestures as parameter entities. To identify each entity, we define their *Persistent Names (PN)*, composed of a set of Persistent Ids to keep track of all gestures that have impacted that entity (see section 3.2.2). More precisely, $PN = \{PI\}.\langle o \rangle$, where $\{PI\}$ is a set of Persistent Ids of the representative darts of the orbit, and $\langle o \rangle$ is the orbit type of the entity.

The parametric specification shown in Figure 3 is: Step 1 : 1-PentagonCreation; Step 2 : 2-EdgeInsertion($PN1$, $PN2$); Step 3 : 3-Triangulation($PN3$); Step 4 : 4-Coloring($PN4$, Yellow), where $PN1$, ..., $PN4$ are respectively the Persistent Names containing the Persistent Ids detailed in Table 1.

PN	PI	O. type	PN	PI	O. type
$PN1$	$\{PI_a\}$	$\langle 1 \rangle$	$PN3$	$\{PI_c\}$	$\langle 01 \rangle$
$PN2$	$\{PI_b\}$	$\langle 1 \rangle$	$PN4$	$\{PI_c\}$	$\langle 01 \rangle$

Table 1: Persistent Ids and orbit types related to gesture parameters of the initial specification.

To illustrate the behaviour of our persistent naming mechanism, we modify the initial specification by adding a Vertex Insertion operation (denoted as A-Step 1) between Step 2 and Step 3 (Figure 4). The reevaluation proceeds as follows.

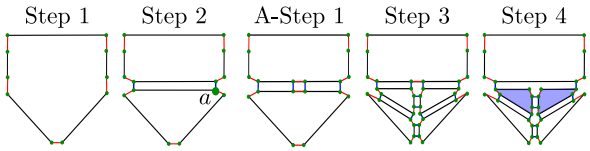


Figure 4: Specification reevaluation.

(1) 1-PentagonCreation is reevaluated the same way as in the initial evaluation (the related rule is applied). (2) 2-EdgeInsertion($PN1$, $PN2$). $PN1$ and $PN2$ will be used to find darts automatically, in order to call the corresponding rule. (3) Add-1-VertexInsertion($a.\langle 02 \rangle$) adds a vertex on edge $a.\langle 02 \rangle$ directly designated by the user during the reevaluation process. (4) 3-Triangulation($PN3$) is not modified. Using $PN3$, we find a dart representing the face and apply the related rule. (5) 4-Coloring($PN4$, Blue) is also reevaluated, finding the darts corresponding to $PN4$ but with a different color parameter. Due to Add-1-VertexInsertion, the initial face has been split, so the new coloring is applied to both sub-faces.

As shown above, determining the types of edition undergone by gestures is mandatory to apply the reevaluation. But to achieve the matching of entities, it is also required to determine how the Persistent Names of referenced orbits have evolved.

3.2 Orbit evolution

We consider the evolution of orbits for both initial evaluation and reevaluation. First, we define the different types of orbital evolutions that may happen (Section 3.2.1). Then, to match evaluation and reevaluation

ISBN 978-80-86943-42-8

entities, we detail the structures of related Ids and Persistent Names (Section 3.2.2). Finally, we propose a structure allowing to follow the entities during the evaluation and a tree structure allowing to report the matching during the reevaluation (Sections 3.2.3 to 3.2.5).

3.2.1 Evolution types

We define the following types of orbit evolution, some of which are shown in Figures 3 and 4. (a) *Creation*: creates a new orbit. (b) *Deletion*: removes an orbit, so no constructive gesture can use it anymore. (c) *Fusion*: merges several orbits. (d) *Modification*: modifies the orbit without any splitting or merging. (e) *NoEffect*: does not affect the orbit. (f) *Split*: splits the orbit.

3.2.2 Persistent naming

The Persistent Id (*PI*) of a dart is set at the time of dart creation, and then modified each time the dart is rewritten by rules. Each *PI* consists of the various operation numbers and rule nodes that have created or rewritten the related dart. For example, dart c of the initial set (Figure 3) is created by instantiating node n_2 of the rule defining 2-EdgeInsertion (Figure 5): "$2-n_2$" is thus a part of PI_c. But n_2 itself is the rewriting of node n_0 located on the left side of the rule, which is associated with dart a in the initial set. Since a has been created by instantiating node n_7 of the rule defining 1-PentagonCreation, PI_c is defined as $\{1-n_7;2-n_2\}$.

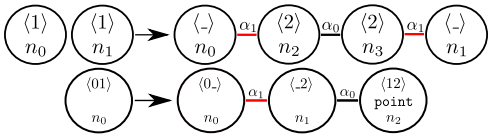


Figure 5: Transformation rules. Top: Edge insertion. Bottom: Triangulation.

The *PN* (see Section 3.1) is used as a parameter of the operations. Thus, 3-Triangulation, which tessellates the face adjacent to dart c, has face Persistent Name $PN3 = \{\{1-n_7;2-n_2\}\}.\langle 01\rangle$ as topological parameter. 4-Coloring is also applied to the face adjacent to c. However, $PN3$ is different from $PN4$ because the face (and therefore c) has been affected by triangulation: $PN4 = \{\{1-n_7;2-n_2;3-n_0\}\}.\langle 01\rangle$.

3.2.3 Rule bulletin boards

Following orbit evolution over several steps of the specification requires to follow evolution depending on each gesture. We use structures called *bulletin boards* for that purpose. Bulletin boards are essential to any monitoring system, but have been very little detailed in the literature.

Our approach is rule-specific: a bulletin board is generated when the user creates a rule to account for the different types of evolution (Section 3.2.1). Figure 6 shows the bulletin board for Vertex Insertion operation. There is one box per orbit type. Inside each box, we describe the evolution types for the rewritten nodes. Let $\langle x\rangle$ be an orbit type: we gather the nodes of the right side of the rule, whose rewriting instantiates darts belonging to the same $\langle x\rangle$, then we search for the left-side nodes which have rewritten these darts, and for which orbit type. A tree is then created for each set: the root contains the nodes selected on the right side, and the leaves contain left-side nodes and the related orbit. The joining arc is labeled with the type of evolution carried out.

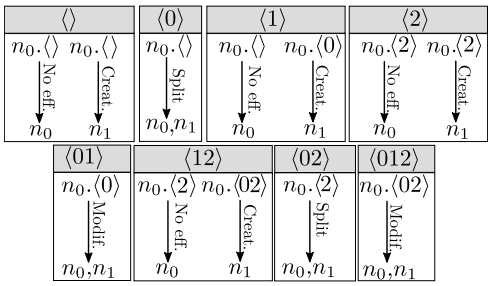


Figure 6: Vertex Insertion bulletin board.

As an example, consider the orbit type $\langle 12\rangle$ in the bulletin board displayed in Figure 6. Figure 7 focuses on vertices (orbit type $\langle 12\rangle$): two vertices are composed of darts instantiated by node n_0 in the right side of the rule whereas the central vertex is made of darts instantiated by node n_1. The two vertices are composed of all darts instantiated from $n_0.\langle 2\rangle$ on the left side of the rule. The type of evolution of these vertices is no effect, because we simply have the same darts in the vertex before and after the rule is applied. A tree is thus created with root labeled "$n_0.\langle 2\rangle$" and leaf labeled "n_0", linked by the "No Eff." arc. The central vertex is composed of all darts instantiated from $n_0.\langle 02\rangle$ on the left side of the rule. This vertex did not exist before the rule is applied. A second tree is thus created, with root labeled "$n_0.\langle 02\rangle$" and leaf labeled "n_1", linked by the "Creation" arc.

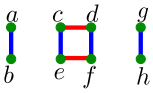


Figure 7: Topological view of edge vertices.

3.2.4 History Record

Bulletin boards are completed by *history records* to process the whole specification. History records analyze the successive bulletin boards of the rules that have impacted any dart. One carries out as many history records as there are *PI*. Let $PN = \{PI_a, PI_b, \ldots\}.\langle x\rangle$ be a Persistent Name. Let $PI_b = \{1-n_i;\ldots;k-n_j\}$ be the Persistent Id of dart b. To create the history record of PI_b,

ISBN 978-80-86943-42-8

we scan its contents in reverse order (from the most recent to the oldest). Therefore, we first consider $\langle x \rangle$ and $k - n_j$ (the last rewriting of dart b by the node n_j of the related rule set at step k). In the bulletin board of this rule, we retrieve the box corresponding to $\langle x \rangle$ and we select the (unique) tree whose child contains n_j. This process is then repeated by going back up each operation constituting PI_b, knowing that it retrieves, for the operation $(k-1)$, the box of the bulletin board corresponding to the orbit indicated at the root of the tree used for operation k.

To illustrate this point, let us create the history record for 4-Coloring applied to $PN4$ (see Figures 3 and 4), that has $\{PI_c\}$ as Persistent Id (see Table 1). The result is shown in Figure 8, with green or red arrows labeling the 6-step process. Before applying 4-Coloring, $PI_c = \{1 - n_7; 2 - n_2; 3 - n_0\}$ and $PN4 = \{PI_c\}.\langle 01 \rangle$, meaning that 4-Coloring is to be applied to orbit $\langle 01 \rangle$ (see the bottom of Figure8(a)). Assume the last element of PI_c (i.e. $3 - n_0$, that is 3-Triangulation applied to n_0) has been initially recovered. Step 1: we look at orbit type $\langle 01 \rangle$ in the Triangulation bulletin board, that is the last rule having impacted c before coloring. At this stage, c is rewritten by node n_0. Step 2: Figure 6 shows that, for orbit type $\langle 01 \rangle$, n_0 results from a split of $n_0.\langle 0 \rangle$. The related excerpt of the Triangulation bulletin board is shown in Figure 8(a).

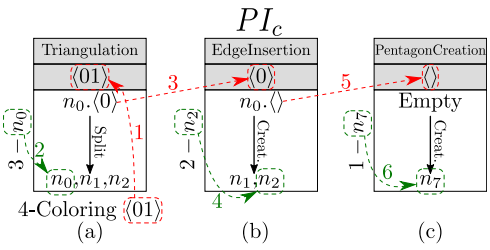


Figure 8: History record of $PN4$.

Step 3: using this orbit type $\langle 0 \rangle$ as an index in the bulletin board of the previous gesture recorded (i.e. 2-EdgeInsertion), we search among the trees related to this entry, the one which contains n_2, for the corresponding identifier in PI_c is $2 - n2$ (Step 4). We find a tree with root $n0.\langle\rangle$ (Figure 8(b)). We repeat the process once again: at Step 5, we go through the bulletin board associated with the previous recorded gesture (1-PentagonCreation). Using the orbit type $\langle\rangle$ as an entry, we search for the related tree which contains n_7, since the corresponding identifier $1 - n7$ (Step 6). The root of this tree has `Empty` as root (see Figure 8(c)), meaning that there is no previous gesture.

The history record of every Persistent Name is carried out in a similar way. As an example, Figures 9 show the history record related to $PN3$.

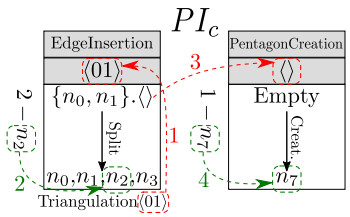


Figure 9: History records of $PN3$.

3.2.5 *Entity matching*

Performing reevaluation requires to match entities between both evaluation and reevaluation specifications. For each history record, a *matching tree* is built, with a Persistent Id as root and orbits as leaves. A matching tree allows to determine which darts of the reevaluation will be used for each orbit designated in the initial set.

For each constructive operation called during reevaluation, we focus on the type of edition which has impacted it. We refer to gestures shown in Figure 4 to describe various scenarios. Considering any gesture already present in the initial evaluation (e.g. gestures 1, 2, 3, 4), matching trees are updated in order to reevaluate this gesture. In case of adding a gesture (e.g., Add-1-VertexInsertion), the bulletin board of the related rule is used to update the matching trees according to the orbits impacted by this addition. In case of deletion, the impacted tree branches are not updated.

We now detail step by step this reevaluation for $PN3$ and $PN4$, as $PN1$ and $PN2$, which are used as parameters of edge insertion gestures, do not involve any particular issue during reevaluation: they use Persistent Ids which have been present since the beginning of the specification and have been impacted by only one gesture.

1-PentagonCreation reevaluation
Since this gesture has no parameter, it is reevaluated in the same way as the initial evaluation. Since the matching trees of $PN1$ to $PN4$ are all impacted by this gesture, they are updated. Figure 10 shows the model after applying the rule, and the impact on the matching trees of $PN3$ and $PN4$. History records shown in Figures 8 and 9 are scanned, one gesture after another, to match darts and orbits in the reevaluated model. Consider $PN3$ for instance: the history record of PI_c indicates that to process 1-PentagonCreation, one must find the newly created orbit type $\langle\rangle$, associated with the instance of node $n7$. A branch of the matching tree is thus created, related to the orbit found in the reevaluated model (a' is the dart instanciated by $n7$). Similarly, one matching tree is generated for $PN4$ using the history record in Figure 8(c).

2-EdgeInsertion reevaluation
Since this operation takes both $PN1$ and $PN2$ as parameters, we use the orbits found in their matching trees.

ISBN 978-80-86943-42-8

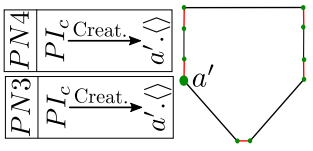


Figure 10: Matching trees and model after the reevaluation of 1-PentagonCreation.

Matching trees of *PN*3 and *PN*4 are updated as in the previous step (as shown in Figures 8 and 9, their respective history record contains the operation prefix "2−"). Figure 11 shows the model after applying the rule, and the impact on matching trees.

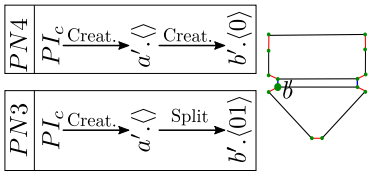


Figure 11: After 2-EdgeInsertion.

Add-1-VertexInsertion addition

This new insertion gesture (relatively to the initial evaluation) requires to trace its impact on orbits currently traced with matching trees.

To determine which parts of the bulletin board related to this rule are relevant, we examine the current leaves of matching trees of *PN*3 and *PN*4 (Figure 11): $b'.\langle 01\rangle$ and $b'.\langle 0\rangle$, resp. The leaves are impacted by this gesture, so we use the trees of the bulletin board of the rule which have $\langle 01\rangle$ and $\langle 0\rangle$ as roots to trace this impact. These trees are displayed in Figure 6. Orbit type $\langle 01\rangle$ undergoes a modification impacting $n0$ and $n1$. The instantiation of any of those nodes can be chosen (here we keep b', i.e. the respective instances of $n0$). Regarding $\langle 0\rangle$, there is a split impacting $n0$ and $n1$. Once again, we choose the instantiation of any node, but since it is an added split, we must trace one dart for each resulting half-edge (we keep both b' and c', i.e. the instance of $n0$ for each half-edge). Matching trees are updated accordingly, as shown in Figure 12.

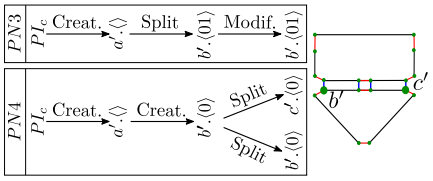


Figure 12: After Add-1-VertexInsertion.

3-Triangulation reevaluation

This gesture is reevaluated on *PN*3. Since *PN*3 is no longer used afterwards, its matching tree can now be removed. Only *PN*4 is impacted by 3-Triangulation (see Figure 8(a)). The result of this reevaluation is displayed in Figure 13.

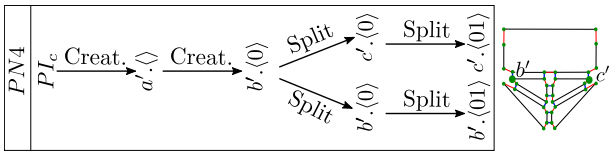


Figure 13: After 3-Triangulation.

4-Coloring reevaluation

Since the color parameter has been modified during reevaluation, 4-Coloring sets the color of the face designated by *PN*4 to blue. Its matching tree indicates that the orbit used in the initial evaluation now corresponds to both $b'.\langle 01\rangle$ and $c'.\langle 01\rangle$ (see Figure 13). Each face is therefore colored. Since *PN*4 is no longer used afterwards, its matching tree is removed.

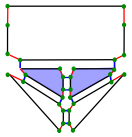


Figure 14: After 4-Coloring.

The final result is shown in Figure 14.

4 CONCLUSION

We propose a new persistent naming system and an entity matching algorithm combining the strong points of graph transformation rules and G-Maps to create and reevaluate new models using parametric specification. Those tools lay the foundation to develop 3D modeling operations dedicated to specific domains such as Archaeology.

Our approach specifically addresses the issue of naming in the context of parametric specification edition (adding and deleting gestures). The naming mechanisms make it possible to name all types of entities in an homogeneous and general way, whatever the dimension of the model. We define unambiguously *Persistent Identifiers* of darts and *Persistent Names* of entities, using the information returned by transformation rules. We follow the evolution of a limited number of entities during both evaluation and reevaluation, in order to achieve matching. Only entities that are actually referenced in the parametric specification are traced, and only during the reevaluation phase. This allows us to hope for space and time savings, but a comparative study will have to be carried out in future works.

To follow entity evolution after applying a gesture, we use the bulletin board associated with the rule defining this gesture. At this time, rule bulletin boards have to

ISBN 978-80-86943-42-8

be designed by the (human) rule designer; it would be interesting to generate them automatically. We also intend to study the effect of changing the order of operations in the parametric specification, since this feature has never been proposed in the literature.

Finally, the full integration of the archaeological dimension into our works will require to study *spatio-temporal reevaluation*, i.e. a reevaluation that, with parameters such as dates, would or would not reevaluate some gestures, in order to give an account of the state of a building through the ages.

5 REFERENCES

[Bab10] Baba-Ali, M. Système de nomination hiérarchique pour les systèmes paramétriques. PhD thesis, 2010. http://nuxeo.edel.univ-poitiers.fr/nuxeo/site/esupversions/16648b94-bfdc-48c0-8ed9-a8d16d5c336c.

[BALB14] Belhaouari, H., Arnould, A., Le Gall, P., and Bellet, T. Jerboa: A Graph Transformation Library for Topology-Based Geometric Modeling, in Int. Conf. on Graph Transformation (ICGT), pp. 269–284, 2014.

[EEPT06] Ehrig, H., Ehrig, K., Prange, U. and Taentzer, G. Fundamentals of Algebraic Graph Transformation, Monographs in Theoretical Computer Science, An EATCS Series Springer, 2006.

[GBS14] Gomes L., Bellon O. P. R., Silva L. 3D reconstruction methods for digital preservation of cultural heritage: A survey. Pattern Recognition Letters 50 (Dec.), pp. 3–14, 2014.

[HMV09] Haegler S., Müller P., Van Gool L. Procedural modeling for digital cultural heritage. Journal on Image and Video Processing - Special issue on image and video processing for cultural heritage, 2009 (Feb).

[JER] http://xlim-sic.labo.univ-poitiers.fr/jerboa/

[Kri95] Kripac, J. A mechanism for persistently naming topological entities in history based parametric solid models. Proc. of the 3rd ACM symposium on Solid Modeling and Applications (SMA'95), pp. 21–30, 1995.

[Lie91] Lienhardt, P. Topological models for boundary representation: a comparison with n-dimensional generalized maps. Computer-Aided Design, 23:1, 1991.

[WN05] Wang, Y. and Nnaji, B.O. Geometry-based semantic id for persistent and interoperable reference in feature-based parametric modeling. Computer Aided Design, vol. 37, pp. 1080–1093, 2005.

[PH89] Prusinkiewicz, P. and Hanan, J. Lindenmayer Systems, Fractals, and Plants, 1989.

[PLH90] Prusinkiewicz, P., Lindenmayer, A. and Hanan, J. The algorithmic beauty of plants, Virtual laboratory, Springer-Verlag, 1990.

[QB15] Quattrini R. and Baleani E. Theoretical background and historical analysis for 3D reconstruction model. Villa Thiene at Cicogna. Journal of Cultural Heritage 16, pp. 119-125, 2015.

[Ver10] Vergnieux R. L'usage scientifique des modéles 3D en archèologie. De la validation Ã la simulation . Proc. of ARQUEOLÓGICA 2.0 Symposium, in Issue 3 of the Int. Journal Virtual Archaeology Review (VAR), 2010.

[Wit13] Wittur J. Computer-Generated 3D-Visualisations in archaeology. Between added value and deception. British Archaeological Reports (BAR) Int. Series, 2013.

[XJHY16] Xue-Yao G., Jia-Qi L, Hao G. and Yun-Feng G. Name and maintain topological faces in rotating and scanning features. Int. Journal of Grid and Distributed Computing, 9:3, pp. 21–26, 2016.

 ISBN 978-80-86943-42-8

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

The Fast Semi-bounded Kernel-Diffeomorphism Estimator

Ibtissem Ben Othman
GRIFT Research Group
CRISTAL Laboratory
School of Computer Sciences
2010, Manouba, Tunisia
ben.othman.ibtissem@gmail.com

Molka Troudi
ECSTRA Laboratory
Department of Quantitative Methods
IHEC Carthage
Carthage 2016, Tunisia
molkaghorbel@gmail.com

Faouzi Ghorbel
GRIFT Research Group
CRISTAL Laboratory
School of Computer Sciences
2010, Manouba, Tunisia
faouzi.ghorbel@ensi.rnu.tn

ABSTRACT

We introduce, by this work, a fast method to estimate probability density functions in the semi-bounded case. This new technique is a simplified version of the kernel-diffeomorphism estimator which requires complexity in the calculations. It is based on a logarithmic transformation of the data which will be estimated by the conventional kernel estimator. Thus, the algorithm complexity is reduced from $O(N^2)$ to $O(N)$.

Keywords

Kernel density estimator, Kernel-diffeomorphism Plug-in algorithm, Fast semi-bounded kernel-diffeomorphism Estimator.

1 INTRODUCTION

The estimation of probability density functions (pdf) is often required to study the complex technological systems and scientific phenomena. The various examples of statistics correspond to distributions with bounded or semi-bounded supports. To estimate the pdf, there are two classes of methods: parametric or non-parametric. In most situations, probability densities are unknown, such operation can be done by the non-parametric methods which are more precise. The histogram method [Silverman86], the orthogonal functions [Hall82] and the kernel method [Fukunaga13] are among the most frequently used non-parametric procedures. The histogram method has the disadvantage of discontinuity. Although the method of orthogonal functions is suitable for any type of support, however it produces the Gibbs effect.

In our research, we have opted for the non-parametric kernel method. In order to ensure a good quality of estimation, it is important to maximize the value of the smoothing parameter by minimizing the mean integrated squared error. The optimization of the bandwidth is performed by the diffeomorphism Plug-in algorithm. A direct resolution of the equation for calculating the optimal value of the smoothing parameter seems very difficult [Saoudi09]. The Plug-in method suggested in [Saoudi09] presents the iterative resolution of its equation. Moreover, a fast variant of this algorithm was developed in [Saoudi09].

In the case of densities with bounded or semi-bounded support, the conventional kernel method is no longer adequate and may present convergence problems at the edges: the Gibbs phenomenon. Several authors have attempted to solve this problem and have presented some methods for estimating distributions with bounded or semi-bounded supports. Among them, we can cite the orthogonal functions [Hall82] and the kernel-diffeomorphism estimator [Saoudi09]. This last method, which is derived from the conventional kernel estimator, is based on a suitable variable change by a C1-diffeomorphism. Inspired by the kernel method, it is important to maximize the value of the smoothing parameter in order to ensure good quality of the estimate. The optimization of the bandwidth is performed by the diffeomorphism Plug-in algorithm, which is a generalization of the conventional Plug-in algorithm [Jain00]. However, its implementation presents additional difficulties compared to the classical version.

In order to surpass the complexity reasons of implementation, we introduce in this work a new variant of the diffeomorphism Plug-in algorithm in the semi-bounded case. This version is based on the logarithmic variable change. The divergence to zero problem of the logarithmic function is solved by the addition of a small strictly positive rate.

The rest of this paper is organized as follows. We start by recalling the kernel method for density estimation in section 2 . The third section presents the contribution of

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.8

the current research which constitutes the fast version of the kernel-diffeomorphism Plug-in algorithm in the semi-bounded case. And in section 4, we have opted for a comparative study between our new variant, the diffeomorphism and the conventional Plug-in algorithms in the semi-bounded case.

2 THE KERNEL ESTIMATE METHOD

The non-parametric Kernel Density Estimator (KDE) has been introduced by Rosenblatt in 1956 [Rosenblatt56] and developed by Parzen in 1962 [Parzen62]. In the case of data with bounded or semi-bounded support , a recent method based on a C1-diffeomorphism variable change has been developed by Saoudi and al. in [Saoudi94, Saoudi97] and called the Kernel-Diffeomorphism density Estimate (KDE). Similarly to the kernel method, the bandwidth optimization is necessary. The Plug-in adapted to this kernel variant is called the Kernel-Diffeomorphism Plug-in (KDP) algorithm.

2.0.1 Kernel-Diffeomorphism Estimator

In this paper, we deal with the semi-bounded densities case. Therefore, a more accurate estimate will be obtained using the KDE which reduces significantly the Gibbs effect [Saoudi94, Saoudi97]. This estimator is a generalization of the KDE. It is suitable for the functions defined on the interval $[a,b]$. The density function is expressed by:

$$\hat{f}_N(x) = \frac{|\Phi'(x)|}{Nh_N} \sum_{i=1}^{N} K(\frac{\Phi(x) - \Phi(X_i)}{h_N}) \qquad (1)$$

where Φ is a C1-diffeomorphism which has the infinity for limit as x approaches a or b. The problematic of optimizing the smoothing parameter can be resolved by using the same methods as those used for conventional kernel analysis. However, as shown in [**?**], an asymptotic study of the KDE, allows better approach to optimal smoothing parameter in the Mean Integrated Squared Error (MISE) sense. Then, its expression becomes the following:

$$h_N^* = [M_\Phi(K)]^{1/5}[J_\Phi(f)]^{-1/5}N^{-1/5} \qquad (2)$$

where $M_\Phi(K) = M(K)\int_R |\Phi'(x)|f(x)dx$ and $J_\Phi(f) = \int_R \frac{F^2(x)}{[\Phi'(x)]^8}dx$

F(x) =[f(x)[$3\Phi''(x)^2 - \Phi'(x)\Phi'''(x)$]] $- 3f'(x)\Phi'(x)\Phi''(x) + f''(x)[\Phi'(x)]^2$
(3)

2.0.2 Kernel-diffeomorphism Plug-in algorithm

The implementation of this extended version presents further difficulties compared to the classical Plug-in algorithm. Indeed, for the conventional Plug-in, $M(K)$ is a constant which can be determined analytically or numerically. As for the KDE adapted plug-in algorithm, $M(K)$ depends on unknown pdf. Similarly, $J(f)$ depends not only on f'', but also on f and f'. Therefore, the complexity of the KDP is increasing. We describe below the kernel-diffeomorphism Plug-in algorithm and its computing complexity:

Step 1: Initialize arbitrary $M_\phi(K)$. In practice $M_\phi^0(K)$ can be equal to $M(K)$.

Step 2: Fix arbitrary $J_\phi^0(f)$, then deduce the first value of the optimal bandwidth; h_N^0. Estimate $f^{(0)}$.

Step 3: Approximate the different quantities: $M_\phi^{(k)}(K)$, $f^{(k)'}$ and $f^{(k)''}$ for each iteration k.

Step 4: Estimate $J_\phi(f^{(k)})$. The value of $h_N^{(k)}$ is so deducted from the k^{th} iteration.

Step 5: Approximate $f^{(k)}$. Stop the algorithm when the difference between $h_N^{(k)}$ and $h_N^{(k-1)}$ is relatively low (below 1%).

Let N be the sample size and p the resolution defined as the point number for which f is estimated. The number of elementary operations is of the order of N^2p which involves a polynomial complexity of $O(N^2)$. Whereas, the conventional Plug-in complexity is linear in the order of $O(N)$.

3 CONTRIBUTION

For simplicity implementation, fast computation and convergence reasons, we introduce, in this section, the Fast Semi-bounded Kernel-Diffeomorphism Estimator (FSKDE). The optimization of the smoothing parameter is performed by the proposed Fast Semi-bounded Kernel-Diffeomorphism Plug-in algorithm (FSKDP). The FSKDP provides a significant improvement in the estimation of the semi-bounded densities. The idea consists on using the logarithmic change of the data qualified by their semi-infinity support: $\Gamma = Log(\zeta)$. Thus, the conventional Plug-in algorithm can now be applied to the transformed data.

3.1 The fast semi-bounded kernel-diffeomorphism estimator

Let's consider independent and identically distributed random variables: $\zeta : \Omega \to R$. Let f_ζ be their probability density function with semi-bounded support: $support(f_\zeta) \subset R$. We recall that the probability densities with semi-bounded support present estimation

 ISBN 978-80-86943-42-8

difficulties due mainly to the Gibbs effect. In order to bypass this divergence limitation at the edge of the semi-bounded interval, we consider a logarithmic variable change for the data: $\Gamma = Log(\zeta)$. The data is then transformed into the following new random variable: $\Gamma : \Omega \to R$. In this case, the distribution function of Γ can be expressed as follows:

$$F_\tau(\Gamma) = P[\Gamma < \tau] = P[Log(\zeta) < \tau] \quad (4)$$

$$F_\tau(\Gamma) = P[\zeta < exp(\Gamma)] = F_\zeta(exp(\Gamma)) \quad (5)$$

The probability density function of Γ can be written as follows:

$$f_\Gamma(\tau) = \frac{dF_\tau(\Gamma)}{d\tau} = f_\zeta(exp(\Gamma))exp(\Gamma) \quad (6)$$

We try to estimate f_ξ as a function of f_Γ. After the change of variable $t = \exp(\Gamma)$, we find f_ζ :

$$f_\zeta(t) = \frac{f_\Gamma(Log(t))}{t} \quad (7)$$

3.2 Convergence problem

The FSKDE presents a specific problem for zero. Indeed, the logarithmic function is defined only on $]0,+\infty[$. It is therefore imperative to submit the data onto a translation before carrying out the logarithmic transformation. This problem is illustrated in figure 1(a). Figure 1(a) represents the density of the exponential distribution $f(x) = \exp(x)$,

The estimation of the exponential density by the FSKDE illustrates the problem described above. Moreover, figure 1(a) shows the divergence at 0 of the estimated exponential distribution.

The idea consists in adding a small positive coefficient ε and thus using the logarithmic variable $Log(x+\varepsilon)$. We notice the resolution of the problem of divergence in 0 as shown in figure 1(b).

3.3 The fast semi-bounded kernel-diffeomorphism Plug-in algorithm

For simplicity implementation, fast computing and convergence reasons, we have introduced a new version of the kernel-diffeomorphism Plug-in algorithm in the case of semi-bounded support. The FSKDP provides a significant improvement in the estimation of the semi-bounded densities. The idea consists on using the logarithm change of the error rates values qualified by their semi-infinity support: $\Gamma = Log(\zeta + \varepsilon)$. Thus, the conventional Plug-in algorithm can now be applied to the transformed data. In order to specify a new classification quality measure, we perform a sequence of three steps:

- **Step 1:** calculate the kernel estimator of the changed variables in the logarithm space: $\Gamma = Log(\zeta + \varepsilon)$
- **Step 2:** iterate the conventional Plug-in algorithm for the transformed data.
- **Step 3:** return to the original space and compute the density kernel estimator: $\hat{f}_\zeta(t) = \frac{\hat{f}_\Gamma(Log(t))}{t}$

The fast diffeomorphism Plug-in algorithm has the same complexity as the conventional one. Thus, the complexity of the FSKDP is linear in the order of $O(N)$.

4 PERFORMANCE STUDY OF THE FAST SEMI-BOUNDED KERNEL-DIFFEOMORPHISM ESTIMATOR

In this part, a comparative study between the KDE and the FSDKE performance is presented in the semi-bounded case. Three semi-bounded distributions are simulated:

- an exponential law of mean 1,
- a first mixture of two laws:
  - a uniform law U(0,1) with a proportion $p_1 = 0.6$.
  - a Gaussian law N(0.8,0.2) with a ratio $p_2 = 0.4$.
- and a second mixture of three laws:
  - a uniform distribution of parameters $U(0,1,5)$ with a proportion $p_1 = 0.4$.
  - a Gaussian law $N(1.3,0.3)$ with a ratio $p_2 = 0.3$.
  - a Gaussian law $N(2.5,0.4)$ with a proportion $p_3 = 0.3$.

Figure 2 below illustrates the theoretical and estimated distributions of the three simulated laws already cited. The estimation is performed using the conventional kernel method. We notice that the estimate overflows from its natural support; the Gibbs phenomenon.

Figure 3 presents the estimation of these probability densities by the KDE. We note that the problem of the Gibbs effect is solved using this method.

Figure 4 presents the estimation of these probability densities by the FSKDP.

The divergence problem in 0 of the FSKDP based on the change of logarithmic variable $Log(x)$ is clearly observed in the first illustration (a) of figure 1. However, we can notice the resolution of this problem by using the logarithmic variable $Log(x+\varepsilon)$. Indeed, the estimation of the probability densities by the FSKDP is almost perfect. These observations are confirmed by the values of the MISE given in the following table. Moreover, the FSKDP has a speed of calculation (having the low execution times) and a better accuracy of estimation (with minimal MISE) with respect to the KDP.

 ISBN 978-80-86943-42-8

5 CONCLUSIONS

To conclude, we have suggested a fast method to estimate probability density functions in the semi-bounded case; the fast semi-bounded kernel-diffeomorphism estimator. This new technique is based on the fast semi-bounded kernel-diffeomorphism Plug-in with a complexity reduced to $O(N)$.

In our future works, we intend to study the case of bounded support distributions. We will also test the new estimators performance in real data.

6 REFERENCES

[Fukunaga13] Fukunaga, K. Introduction to statistical pattern recognition. Academic Press, 2013.

[Hall82] Hall, P. Comparison of two orthogonal series methods of estimating a density and its derivatives on an interval. Journal of multivariate analysis, 12, pp. 432-449, 1982.

[Silverman86] B. W. Silverman. Density estimation for statistics and data analysis. Chapman and Hall, London; New York, 1986.

[Jain00] Jain, A.K., Duin, R.P.W. and Mao, J. Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence, 22, pp. 4-37, 2000.

[Rosenblatt56] Rosenblatt, M., et al. Remarks on some non-parametric estimates of a density function. The Annals of Mathematical Statistics, 27, pp. 832-837, 1956.

[Saoudi97] Saoudi, S., Ghorbel, F. and Hillion, A. Some statistical properties of the kernel-diffeomorphism estimator. Applied Stochastic Models in Busines and Industry 13, pp. 39-58, 1997.

[Saoudi94] Saoudi, S., Hillion, A. and Ghorbel, F. Non-parametric probability density function estimation on a bounded support: Applications to shape classification and speech coding. Applied Stochastic Models in Business and Industry, 10, pp. 215-231, 1994.

[Saoudi09] Saoudi, S. and Troudi, M., Ghorbel, F. An iterative soft bit error rate estimation of any digital communication systems using a non-parametric probability density function. EURASIP Journal on Wireless Communications and Networking, 4, 2009.

[Parzen62] Parzen, Emanuel. On estimation of a probability density function and mode. The annals of mathematical statistics, JSTOR, 33(3), pp. 1065-1076, 1962.

 ISBN 978-80-86943-42-8

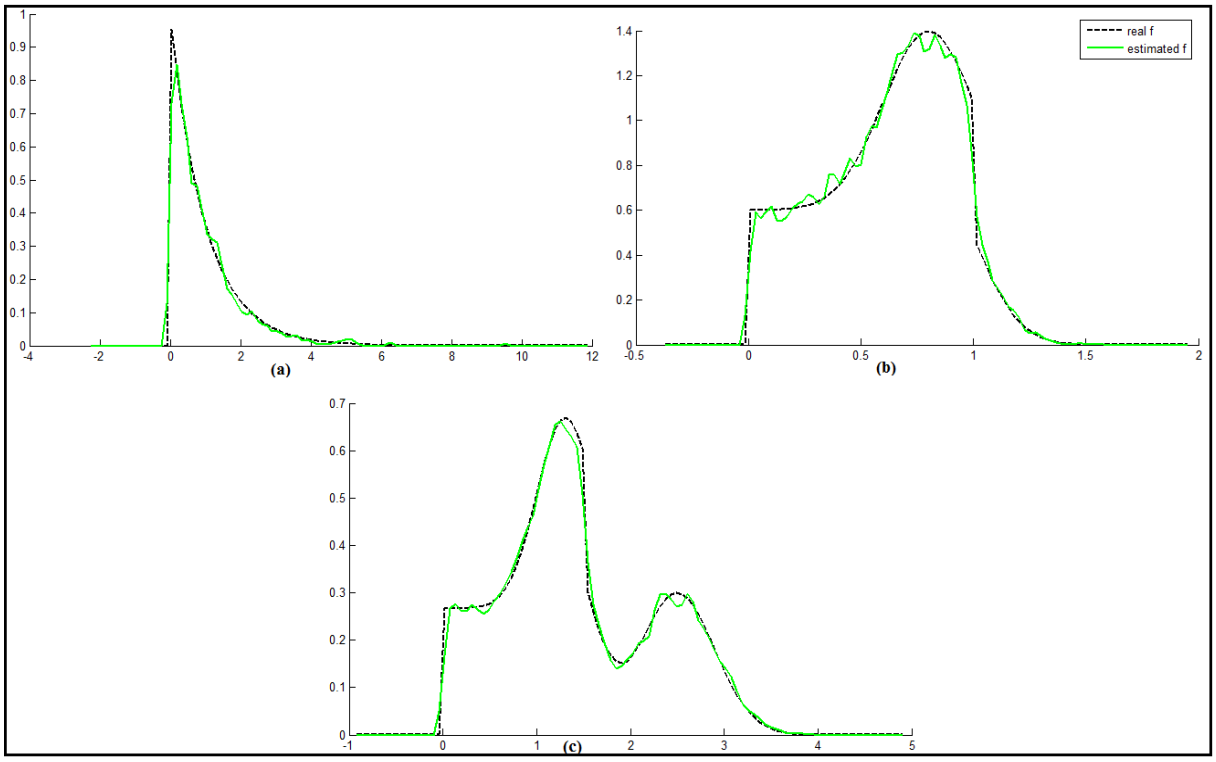


Figure 1: Estimation of the three distributions by the conventional kernel method: (a) the exponential distribution $\exp(x)$, (b) the first mixture and (c) the second mixture.

Method	MISEx10^{-5}	Execution time
KDP	0.1909	0.1248
FSKDP	0.1415	0.0624

Table 1: MISE and execution time of the conventional (KDP) and fast (FSKDP) kernel methods for the estimation of the exponential distribution $\exp(x)$.

Method	MISEx10^{-5}	Execution time
KDP	3.7218	1.0608
FSKDP	2.6252	0.7332

Table 2: MISE and execution time of the conventional (KDP) and fast (FSKDP) kernel methods for the estimation of the first mixture of the uniform law $U(0,1)$ and the Gaussian law $N(0.8,0.2)$.

Method	MISEx10^{-5}	Execution time
KDP	0.4739	1.0764
FSKDP	0.4027	0.7800

Table 3: MISE and execution time of the conventional (KDP) and fast (FSKDP) kernel methods for the estimation of the second mixture of the uniform law $U(0,1,5)$ and two Gaussian laws $N(1,3,0,3)$ and $N(2,5,0,4)$.

ISBN 978-80-86943-42-8

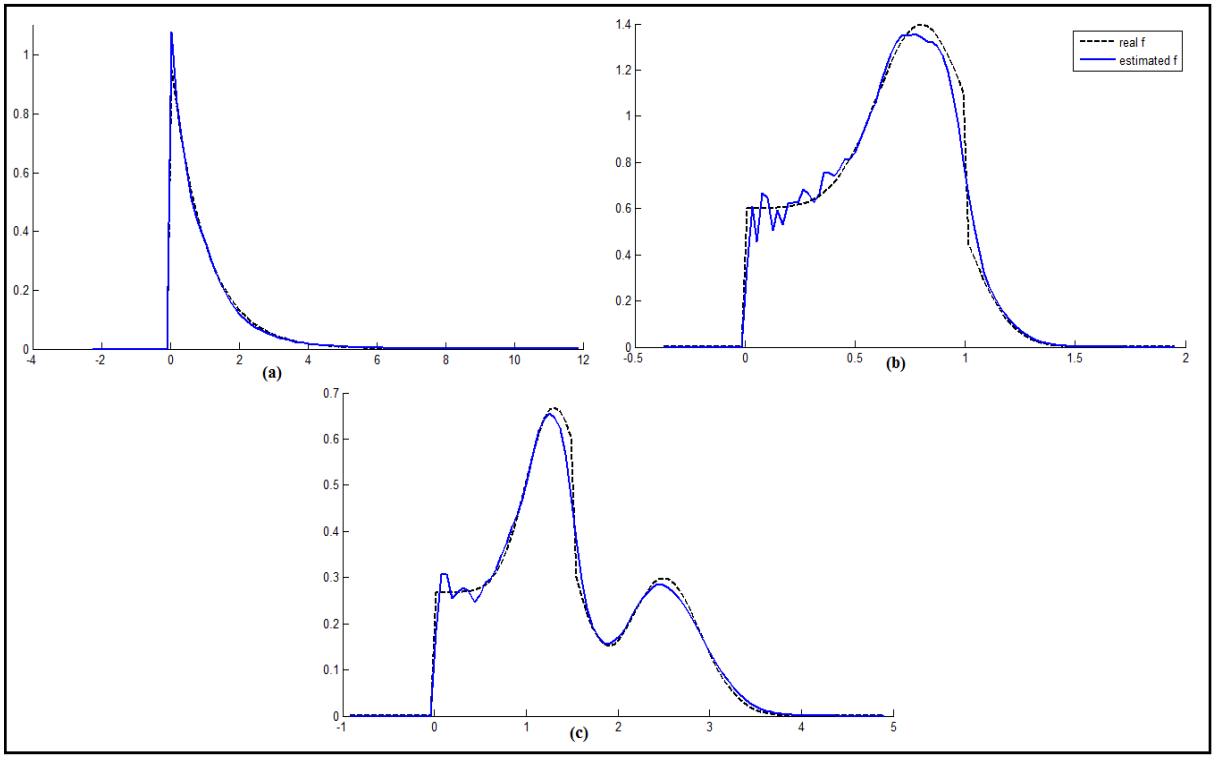


Figure 2: Estimation of the three distributions by the KDP: (a) the exponential distribution $\exp(x)$, (b) the first mixture and (c) the second mixture.

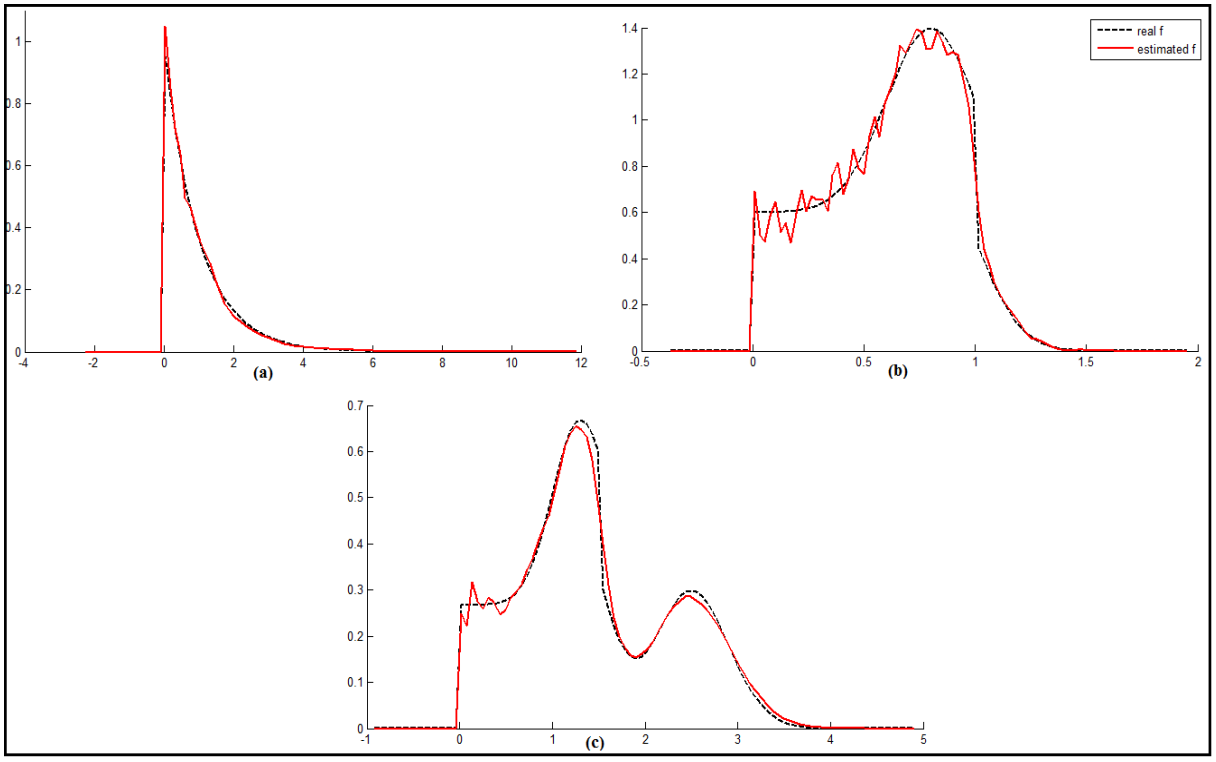


Figure 3: Estimation of the three distributions by the FSKDP: (a) the exponential distribution $\exp(x)$, (b) the first mixture and (c) the second mixture.

ISBN 978-80-86943-42-8

ISSN 2464-4617(print)
ISSN 2464-4625(CD)
Computer Science Research Notes
CSRN 2803
Poster's Proceedings
http://www.WSCG.eu

Using a spatiotemporal plane to recover a moving object's shape using Spatiotemporal Boundary Formation

Douglas W. Cunningham
BTU Cottbus-Senftenberg
Konrad-Wachsmann-Allee 5,
03046 Cottbus, Germany
Douglas.Cunningham@b-tu.de

Mikhail Ashkerov
BTU Cottbus-Senftenberg
Konrad-Wachsmann-Allee 5,
03046 Cottbus, Germany
Mikhail.Ashkerov@b-tu.de

ABSTRACT

When ever an object moves, it successively covers and uncovers surfaces that are farther away. This occlusion and dis-occlusion always occurs precisely at the boundaries of the moving object and as such provide information not only about the shape of the object but also about its velocity, transparency, and relative depth. Humans can and do use this information, and the process has come to be called Spatiotemporal Boundary Formation (SBF). Previous authors have used the wealth of experimental investigations into SBF to create a mathematical model of the process. In this article we proposed a novel method to recover the orientation and velocity the local edge segments of the moving objects which is more flexible, more robust, more compact, and allows the recovery of edges that do not have a constant velocity. The method can be used in object segmentation algorithms or as a pre-filter for machine-learning-based recognition algorithms in order to improve the overall result.

Keywords

Motion perception, Dynamic occlusion, SBF

1 INTRODUCTION

The human visual system is amazingly flexible and powerful. It is not surprising, then, that many computer vision algorithms strive to mimic human capabilities in order to automatically accomplish a wide variety of importance, practical applications in engineering. Many of the human visual system's abilities require that the visual scene first be segmented into component objects. Although most studies in both psychology and computer science focus on the information available from static images, there is a considerable amount of information available over time for object segmentation.

One of the most promising human abilities for object segmentation using dynamic information is Spatiotemporal Boundary Formation (SBF) [1], which is based on the dynamic occlusion of texture caused by the relative motion of two overlapping surfaces. Since our everyday environment is generally cluttered [2], the motion of any given object will successively cover and uncover the background. Interestingly, the background texture will always disappear at the front of the moving objects and will reappear at the back of the moving object. In fact, the texture will appear and disappear precisely at the edges of the object, and therefore provide direct information about many different object properties.

Several researchers have shown that humans can use dynamic occlusion to segment an object from its background, as well as perceive the object's exact shape, velocity, relative depth, and transparency [3, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. A review of this and related phenomena can be found in [21]. In addition to the examination of dynamic boundaries in humans, several researchers have examined the impact of dynamic figure information on the control of fly behavior[22, 23].

One traditional demonstration of dynamically defined edges comes from Michotte's classic tunneling effect [24, 25], where a luminance defined surface is progressively deleted (see Figure 1 for a sketch). As the black circle is displaced upwards, less of it is visible. Perceptually, this is often seen as a complete circle that is being progressively hidden by something that is the same color as the background. While this effect is most convincing in its traditional dynamic form, it can even be seen, to some degree, statically.

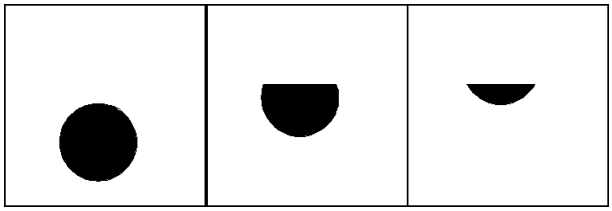


Figure 1: Three frames illustrating Michotte's tunneling effect.

The dynamic transformation of the statically visible surface provides information about both its continued existence and the presence of an occluding form. Thus, two surfaces are seen, one of which has the same color

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.9

as the background and is slowly covering the other one. Gibson and colleagues [2] asserted that the changes over time at the boundaries of an object are sufficient to define the shape of those boundaries. Gibson's (1968) film [26], constructed to support this claim, depicts a square with a speckled texture moving over a similarly textured surface. Since there is no static information for the form or its boundaries (e.g., no global difference in luminance, hue, texture, or depth between the inside and outside of the square), neither the surface nor the boundaries are visible when the figure is stationary. When the figure moves, however, a well-defined surface bounded by clear edges is seen moving in front of another surface with similar texture. The displays is reminiscent of naturally camouflaged animals, which can become nearly invisible when stationary in front of the proper background but are nearly always visible when moving. The edges seen in Gibson's displays, as well those seen in Michotte's demonstration, have a phenomenal quality similar to the edges of illusory figures such as the Kanizsa triangle (see Figure 2), they are strong, sharp, and clearly visible just as normal, luminance defined edges [27].

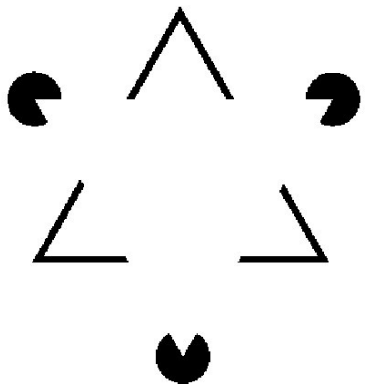


Figure 2: The Kanizsa triangle.

Shipley and Kellman [28] provided a mathematical proof that when the spatial and temporal location of at least 3 elements transformations (e.g., occluded or dis-occluded background elements) are known, we can recover the orientation and velocity of the edge that transformed those elements. Briefly, the spatial and temporal separation between pairs of texture element transformations is represented by local motion vectors. Vector subtraction of the local motion vectors resulting from three non-(spatially)-collinear texture elements defines the orientation of the edge that transformed them. Cunningham and colleagues subsequently modified and extended this model to use a relative encoding scheme (avoiding the explicit requirement for motion vectors) and then modeled the extraction of the occluding form's global shape and its velocity (a preliminary version can be found in [13]). Computational simulations of the model show that it captures the major psychophysical aspects of SBF, including a dependency on the spatiotemporal density of element changes and a sensitivity to spurious changes [11].

Most modern object recognition tasks either require that (or strongly benefit when) the input images are segmented into discrete objects. This is especially true of machine learning algorithms in general and neural networks in particular. Thus, we propose that prefiltering of the input data with SBF can not only segment and make explicit the information that the object recognition algorithms need (i.e., objects) but also significantly decrease the dimensionality of the input and thus will make machine learning algorithms more robust.

2 MATERIAL AND METHODS

The starting point of our algorithm is the same as the previous models of SBF [28]. Specifically, the dataset consists of the set of (background) texture elements $p_i = (x_i, y_i, t_i)$ that have been covered or uncovered by a moving shape. The spatial location of the element (x_i, y_i) is a projection of the elements true three dimensional location onto the observer's fronto-parallel plane (e.g., a camera's sensor array or a human's retina). The time t_i is the time when the point was transformed (either removed or revealed). As with the previous works, we will begin with the following assumptions, derived from human psychophysical experiments:

1. The local edge can be approximated with a straight line.
2. The velocity of the line is constant.
3. The line does not change orientation.
4. The points are not collinear.

It is critical to note that these assumptions are local in both space and time and are flexible. For example, an circle can be approximated by a series of straight edges, and the smaller the straight edges are, the better the approximation will be. Likewise, the velocity and edge orientation only need to be constant for at most 150 ms [21].

Here, we provide a new approach to extracting the local segment's orientation and velocity. This new approach represents the motion of the moving segment as a plane in space-time, with the occluded points lying on – and defining – this plane. This method is more flexible than previous versions, can be extended to use progressive shape segmentation methods as well as to recover shapes with non-constant velocities. The input to this stage and the output from it will be the same as for the two previous models. This means, among other things, that this step can replace the same stage in existing SBF implementations to recover the full shape and global velocity.

ISBN 978-80-86943-42-8

For the sake of simplicity, we will start by defining the algorithm using the smallest possible dataset: three spatiotemporal occlusion events:

$$P = \{p_1, p_2, p_3\} \tag{1}$$

although all the texture appearances and disappearances in the full dataset feed into the SBF algorithm. Indeed, increasing the number of points used to calculate the spatiotemporal edge plane will increase the spatial and temporal accuracy of the results [21].

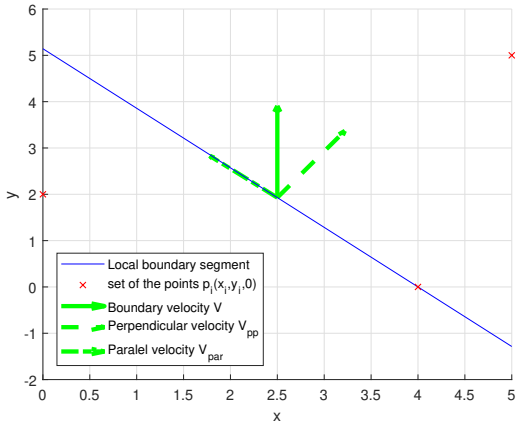


Figure 3: Input data for the method and task definition in projection on XY plane.

Figure 3 depicts the set of points P, the local segment S of the moving edge, and the velocity vector $\overline{V}$. As stated in assumption 1 above, S is represented as an infinite line. Obviously, the edge is a mere segment, but we cannot at any given time directly see the ends of the line. Indeed, at no time do we even see the line, only its effects (occlusion and dis-occlusion). One consequence of this is that additional information is needed to recover the true length of the segment. We will follow the length calculations presented in [11] (essentially, the points in P are projected along the velocity vector of the edge). Another, more critical, consequence of not being able to see the edge segment is that it is not possible *in principle* to recover the component of the velocity that is parallel to the line. This is because is it not possible to uniquely determine any single point on the line at two different points in time. This problem is called the "aperture problem" and was first described in 1935 by Hans Wallach [29]. Thus, we will decompose the velocity vector $\overline{V}$ of the edge into a vector along the boundary $\overline{V_{\parallel}}$ and one perpendicular to the boundary $\overline{V_{\perp}}$ (see figure 3). Our task is to find the velocity perpendicular to the boundary line $\overline{V_{\perp}}$ and the orientation of the local segment of the boundary. The recovery of at least two different edges is sufficient to restore the true velocity $\overline{V}$ of the object [12].

The first step in recovering the orientation of the edge is to realize that each point in P lies on the edge at the instant that is occluded or revealed. Imagine the simplest case where all three points disappear at the same time. Since the all have the same time $t_i = 0$, all the points will be on the fronto-parallel (or XY) plane at the same time, and as such the edge segment will be uniquely defined (as the connection of the three points) and will also lie on the XY plane. Note that technically these events are "collinear in time", and as such we cannot recover the velocity component.

Using a somewhat more complicated case, let's assume that two of three elements changing state at the same time t_1, and the third changes at some later time t_2. We can still plot all three points on the XY plane (that is, show each point where it is on the sensor array, regardless of the time at which it is revealed or deleted). As with the previous case, the local edge segment can be recovered – by definition – as a line connecting the two dots that change together (see Figure 4). As the local edge segment moves, it will occlude the final point at time t_2 and yet will still be parallel to its previous orientation at time t_1. If we were to project the location $(x_i.y_i)$ of the points that disappear at time t_1 along velocity vector perpendicular to the edge $\overline{V_{\perp}}$ to time t_2, it is clear that the projected points will still lie on the edge segment. In other words, if we can "discount" the motion of the edge (such as by projecting the position of the element changes along the velocity vector), we can recover the orientation of the edge segment. Without loss of generality, then, we can allow the three points to change at any location in space and time and they will still lie on a unique plane in space-time (as long as the points are not collinear). The general case, where the three element changes occur at different points in time is shown in Figure 5.

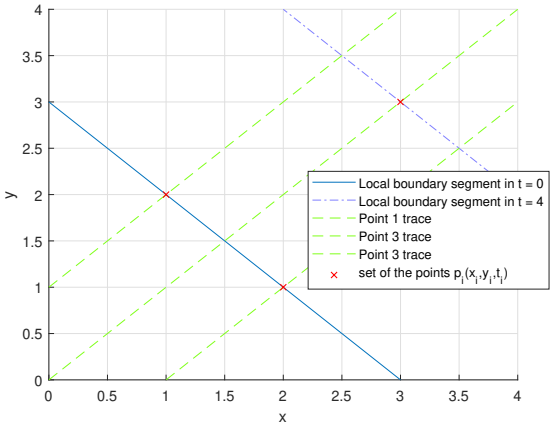


Figure 4: Case with simplifications.

In the next step, we place set of points in a space-time plot by plotting the time t along the z axis (and thus orthogonal to the XY plane). Since the edge will have a spatial extent at any given instant, it clearly must be parallel to the XY plane at every given instant. Like-

ISBN 978-80-86943-42-8

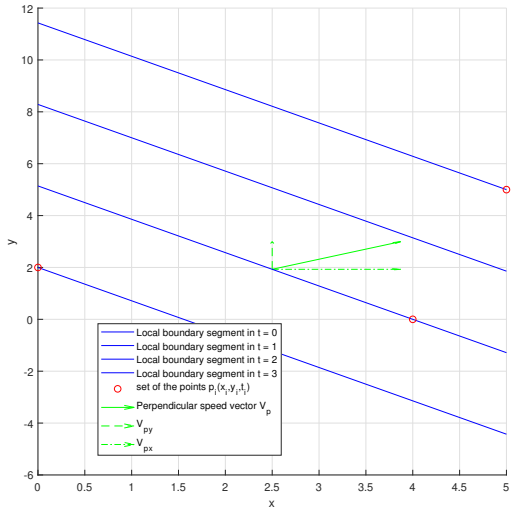


Figure 5: Representation of local borders in time.

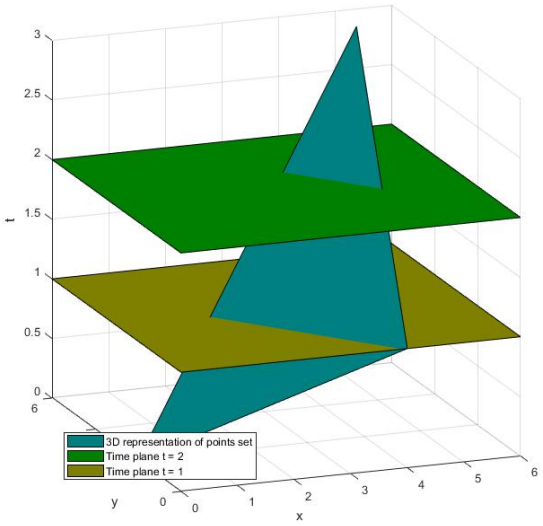


Figure 6: 3D representation of local border.

wise, the edge segment will intersect each point p_i at the correct time t_i. Thus, we can construct through these 3 space-time points a plane P_{set} (from the definition of the plane it can be done uniquely) and the edge segment will lie on this plane. In fact, the edge segment will be the intersection of the 3D space-time plane and the XY plane for any given time t_i (see Figure 6).

To recover the edge orientation, we first write the equation of the plane in the canonical form 2

$$Ax + By + Ct + D = 0 \tag{2}$$

If we solve plane equation with our set of points 1 we obtain the coefficients A, B, C, D for constructing the plane 2 in 3D space. The line of intersection of this plane with the time plane corresponds to the local boundary at the time t_i and its projection onto the XY plane corresponds to the orientation of the segment in two-dimensional space.

Changing the time and finding the intersection of the new P_{time} we get a family of parallel lines on the plane XY that correspond to the local boundary at different times, see (Figure 5).

In the final step, we use the recovered orientation of the line at different times to extract the perpendicular component of the velocity vector $\overline{V_\perp}$. Geometrically, the perpendicular velocity will be equal to the slope of the plane in the direction of the perpendicular to the local segment. This can easily be calculated from the equation of the plane and the local segment. For convenience, we decompose the speed $\overline{V_\perp}$ to the velocity along the axis X and the axis Y:

$$\overline{V_\perp} = \{\frac{-AC}{A^2+B^2}; \frac{-BC}{A^2+B^2}\} \tag{3}$$

Numerically, this method gives us exactly the same solution as the vector and trigonometric methods. Since it takes the same input and provides the same form of output, it can be inserted in the an existing SBF pipeline to recover the gloval shape and velocity. One of the main advantages of this new approach is we can change P_{set} to a quadratic function. We will still be able to recover the orientation and speed, but no longer need to assumption a constant velocity.

3 RESULTS

To test the quality of the edge reconstruction, we created a test film similar to that used in previous works. The film was 300 frames long (at 640x480 pixels) and consisted of a black rhombus moving over a field of white dots on a black background. At no given instant can the edges of the rhombus be seen (i.e., the figure is not defined in the static luminance domain). The Rombus moved from left to the right with a constant velocity. In the figure 7 you can see frames from the film and the object shape recovered with our method. Notice that since there is no surface texture and nothing in the luminance domain actually moves, optic flow algorithms will have extreme difficulties with this display. When compared to the results of previous work (such as [11]), we see that the current method produces slightly cleaner shapes, but has the same problems in corners (see Figure 8).

In sum, we present a method for representing the shape and velocity of a moving shape with a space-time plane. The plane is recovered from the position in space and time of background elements transformed by the edges of the moving shape. The new method produces results that are more accurate than previous models and

ISBN 978-80-86943-42-8

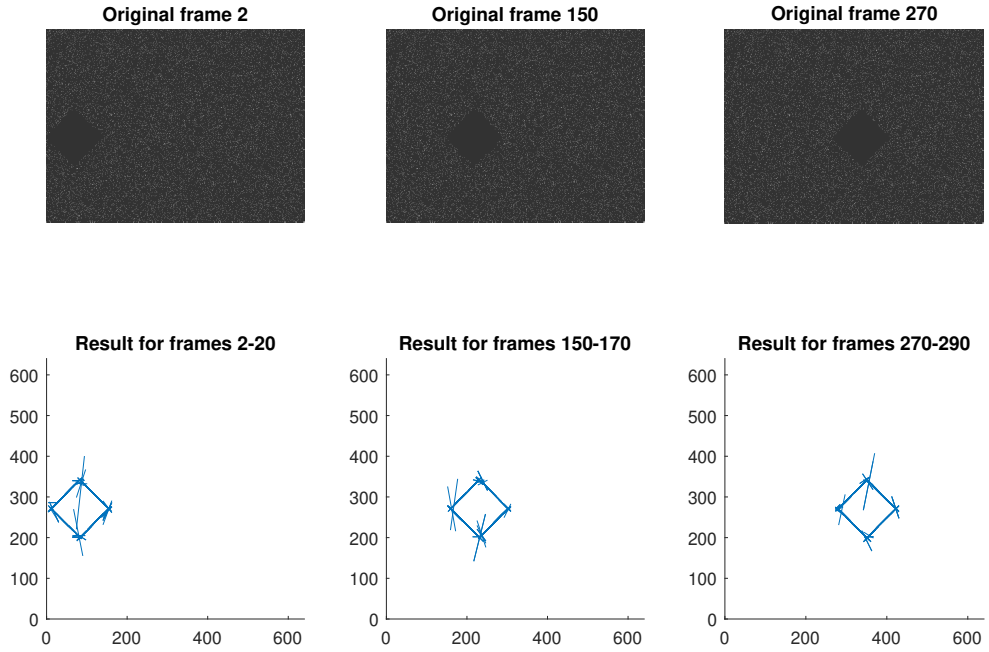


Figure 7: Reconstruction of movement.

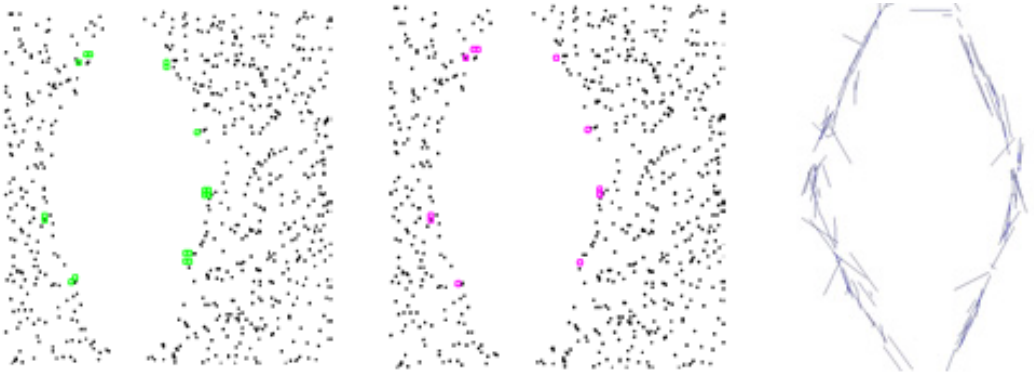


Figure 8: Reconstruction of object shape from poster T.Cooke [11] .

provides a few additional advantages. The new approach produces an infinite family of lines, with line corresponding to an instant in time t from the interval $t \in [t_1 \cdot t_3]$. If we take the time outside of the specified area, we get a linear extrapolation of the boundary motion. This form of the solution is convenient for cluster analysis, allowing us to find similar segments of the object. Likewise, the method is convenient for applying machine learning on sets of initial points, which would improve their grouping in a segment. Finally, the method is easy to modify to remove the assumption that the segment velocity is linear.

4 ACKNOWLEDGEMENTS

The authors would like to thank Stefan Guthe for his suggestions and valuable comments.

5 REFERENCES

[1] T. F. Shipley and P. J. Kellman, "Optical tearing in spatiotemporal boundary formation: When do local element motions produce boundaries, form, and global motion?," *Spatial Vision*, vol. 7, no. 4, pp. 323–339, 1993.

[2] J. J. Gibson, *The ecological approach to visual perception*. Psychology Press, 2013.

[3] G. Erlikhman and P. J. Kellman, "Modeling spatiotemporal boundary formation," *Vision research*, vol. 126, pp. 131–142, 2016.

[4] G. J. Andersen and J. M. Cortese, "2-d contour perception resulting from kinetic occlusion," *Perception & Psychophysics*, vol. 46, no. 1, pp. 49–55, 1989.

ISBN 978-80-86943-42-8

[5] D. R. Bradley and K. Lee, "Animated subjective contours," *Attention, Perception, & Psychophysics*, vol. 32, no. 4, pp. 393–395, 1982.

[6] N. Bruno, M. Bertamini, and F. Domini, "Amodal completion of partly occluded surfaces: Is there a mosaic stage?," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 5, p. 1412, 1997.

[7] N. Bruno and M. Bertamini, "Identifying contours from occlusion events," *Perception & Psychophysics*, vol. 48, no. 4, pp. 331–342, 1990.

[8] N. Bruno and W. Gerbino, "Illusory figures based on local kinematics," *Perception*, vol. 20, no. 2, pp. 259–274, 1991.

[9] C. M. Cicerone and D. D. Hoffman, "Color from motion: dichoptic activation and a possible role in breaking camouflage," *Perception*, vol. 26, no. 11, pp. 1367–1380, 1997.

[10] C. M. Cicerone, D. D. Hoffman, P. D. Gowdy, and J. S. Kim, "The perception of color from motion," *Perception & Psychophysics*, vol. 57, no. 6, pp. 761–777, 1995.

[11] T. Cooke, D. W. Cunningham, C. Wallraven, and H. H. Bülthoff, "Local processing in spatiotemporal boundary formation," in *Proceedings of the 7th Tübingen Perception Conference*, p. 65, Knirsch Verlag Kirchentellinsfurt, 2004.

[12] D. W. Cunningham, T. F. Shipley, and P. J. Kellman, "Interactions between spatial and spatiotemporal information in spatiotemporal boundary formation," *Perception & psychophysics*, vol. 60, no. 5, pp. 839–851, 1998.

[13] D. Cunningham, A. Graf, and H. Bülthoff, "A relative encoding model of spatiotemporal boundary formation," in *5. Tübinger Wahrnehmungskonferenz (TWK 2002)*, 2002.

[14] T. Hine, "Subjective contours produced purely by dynamic occlusion of sparse-points array," *Bulletin of the Psychonomic Society*, vol. 25, no. 3, pp. 182–184, 1987.

[15] P. J. Kellman and M. H. Cohen, "Kinetic subjective contours," *Perception & Psychophysics*, vol. 35, no. 3, pp. 237–244, 1984.

[16] K. Prazdny, "Illusory contours from inducers defined solely by spatiotemporal correlation," *Perception & Psychophysics*, vol. 39, no. 3, pp. 175–178, 1986.

[17] I. Rock and F. Halper, "Form perception without a retinal image," *The American journal of psychology*, vol. 82, no. 4, pp. 425–440, 1969.

[18] T. F. Shipley and P. J. Kellman, "Spatiotemporal boundary formation: Boundary, form, and motion perception from transformations of surface elements.," *Journal of Experimental Psychology: General*, vol. 123, no. 1, p. 3, 1994.

[19] T. F. Shipley and P. J. Kellman, "Spatio-temporal boundary formation: The role of local motion signals in boundary perception," *Vision Research*, vol. 37, no. 10, pp. 1281–1293, 1997.

[20] P. Stappers, "Forms can be recognized from dynamic occlusion alone," *Perceptual and Motor Skills*, vol. 68, no. 1, pp. 243–251, 1989.

[21] T. F. Shipley and D. W. Cunningham, "Perception of occluding and occluded objects over time: Spatiotemporal segmentation and unit formation.," 2001.

[22] H. Bülthoff, "Figure-ground discrimination in the visual system ofdrosophila melanogaster," *Biological Cybernetics*, vol. 41, no. 2, pp. 139–145, 1981.

[23] W. Reichardt and T. Poggio, "Figure-ground discrimination by relative movement in the visual system of the fly," *Biological Cybernetics*, vol. 35, no. 2, pp. 81–100, 1979.

[24] A. Michotte, *On phenomenal permanence: facts and theories*, vol. 7. Acta Psychologica, 1950.

[25] A. Michotte, G. Thine, G. Crabbé, *et al.*, *Les complements amodaux des structures perceptives*. Institut de psychologie de l'Université de Louvain, 1964.

[26] J. Gibson, "The change from visible to invisible: A study of optical transitions [film]," *State College, PA: Psychological cinema register*, 1968.

[27] G. Kanizsa, *Organization in vision: Essays on Gestalt perception*. Praeger Publishers, 1979.

[28] T. F. Shipley and P. J. Kellman, "Spatio-temporal boundary formation: the role of local motion signals in boundary perception," *Vision Research*, vol. 37, no. 10, pp. 1281 – 1293, 1997.

[29] H. Wallach, "Über visuell wahrgenommene bewegungsrichtung.," *Psychologische Forschung*, pp. 325–380, 1935.

 ISBN 978-80-86943-42-8

The solution of the problem of simplifying the images for the subsequent minimization of the image bit depth

Evgenii Semenishchev
Don State Technical University
344000, Russia, Rostov-on-Don
sea.sea@mail.ru

Viacheslav Voronin
Don State Technical University
344000, Russia, Rostov-on-Don
Voronin_sl@mail.ru

Igor Shraifel
Don State Technical University
344000, Russia, Rostov-on-Don
shraifel17@mail.ru

Irina Tolstova
Don State Technical University
344000, Russia, Rostov-on-Don
irinatolstova@yandex.ru

Dmitrii Chernishchov
Don State Technical University
344000, Russia, Rostov-on-Don
dimcher@inbox.ru

ABSTRACT

In this paper, the approach of changing bit depth of images is considered. This type of operation is required when performing primary processing operations, identifying parameters and stitching images. The process of changing bits depth of images is performed in three stages. At each stage, the error minimization criterion is tested Result of applying the approach allows obtaining numerical region characteristics including the number of clusters, the number of minimum and maximum cluster sizes. To perform the process of minimizing some of the criteria, it is necessary to divide the image into areas. The paper presents a mathematical description of the approach, as well as flowcharts for performing operations of data processing steps. The article gives recommendations for choosing coefficients to obtain optimal minimizing parameters. The test images give an example of performing bit changes on image areas.

Keywords

Images, bit, pixel brightness, areas detection, changes brightness.

1. INTRODUCTION

Analysis of streaming video, as well as data, requires preliminary preparation. Analyzing preliminary image, and searching objects on them, will be faster if the frame bit size is reduced. To reduce the data (bits) it is necessary to change the bits depth. The bits depth cannot be changed by combining the nearest values, but it is possible to do it by applying the block generation procedure and averaging the values in them. The application of this approach allows performing maximum changes to stationary areas and leave unchanged some areas the users are interested in . The use of bits changing operation is applicable: to the image area, to the block next to a certain part of the histogram or the whole image. The application of this approach is important for combining data obtained in different electromagnetic ranges. Also it is significant for a boundary layer formation on a low-level image. An example is the integration of IR and optical image data. In this task, it is possible to perform bit depth reduction to the level of the IR sensor.

This direction is relevant for solving the problem of combining data obtained by sensors operating in different electromagnetic ranges [1]. An example is a problem of obtaining an image with the boundaries of objects obtained with a two video camera [2]. Obtaining the boundaries of objects is carried out by analyzing the data obtained with an optical system. When obtaining the boundaries of objects obtained by IR and optical sensors, in conditions of poor visibility [3]. When solving the problem of obtaining depth maps for a set of data obtained from a group of cameras [4]. For preliminary simplification of data, the main task is to improve the performance of systems working with a great amount of data [5]. For preliminary analysis in case of solving the problem of stitching images into one composition [6] and many other problems.

2. STATEMENT OF THE PROBLEM

Let some (first) image be divided into small fragments of the $K_1, K_2, ..., K_n$, and the *i*-th fragment consisting of l_i pixels. Next, the operation of inducing the fragments brightness by the first image

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.10

is to be performed. In this paper they are called– large pixels. For the case of $l_i = 1$, the large pixel K_i is an ordinary pixel. All pixels are known to be equal in the image . The brightness of a large pixel is K_i ,so it is advisable to calculate the arithmetic mean of a_i the brightness of usual pixels is $i = 1,2,...,n$. The image thus obtained will be called the second image, and the set of larger pixels will be denoted by the symbol $\Re$ .The problem is to divide the set into clusters means to give all large pixels of each cluster T_j new luminance value. Averaging of their brightness is performed in accordance to the following two formulas:

$$\beta_j = (\max_{K_i \in T_j} a_i + \min_{K_i \in T_j} a_i)/2, \quad (1)$$

$$\gamma_j = \sum_{K_i \in T_j} a_i l_i \Big/ \sum_{K_i \in T_j} l_i \quad (2)$$

In future, the word combination "splitting into clusters" is reduced to one word "splitting". In case of using the method, all n of large pixels are replaced by brightness of β_j in the first case or γ_j in the second, two more images are obtained, they are called the third and fourth, respectively. Obviously, the necessary condition for quality of the above pointed division is the possibility of enumerating clusters as inequalities

$$\min_{K_i \in T_j} a_i \geq \max_{K_i \in T_{j+1}} a_i, \quad j = 1,2,...,m-1; \quad (3)$$

here: m – is the number of clusters, T_j – is the cluster with the number j. Further dividing the set $\Re$ into clusters, decompositions are pointed out corresponding to the requirement (3). Any decomposition needs implication

$$1 \leq i < j \leq n, \quad a_i = a_j, \quad K_j \in T_\rho, K_j \in T_q \Rightarrow p \leq q \quad (4)$$

In paper we propose an algorithm of partitioning. Its use provides a predetermined degree of proximity of the second and third or (and) second and fourth images. Replacing the first image with the subsequent one allows to reduce the processing time. The problem of choosing between bit depth and image quality is solved.

3. NOTATION AND DEFINITIONS

Let $x = (x_1, x_2, ..., x_m)$ – be any vector with real coordinates. the vector values are arranged in decreasing order to obtain a series of numbers $\tilde{x}_1 \geq \tilde{x}_2 \geq ... \geq \tilde{x}_m$ and, respectively, the vector $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m)$ of the same dimension m.

The calculation procedure of the algorithm presented in the work begins with the construction of the brightness vector of all large pixels $\tilde{a} = (\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_n)$, situated in descending order. For any pair of large pixels $K_i, K_j (i < j)$ with the same luminosity, it is assumed that in the coordinate recording the vector $\tilde{a}$, the brightness of K_i precedes the brightness of K_j. This requirement, together with the monotonicity of the sequence $\tilde{a}$, will ensure, (3), (4), the one-to-one correspondence between all partitions of the set $\Re$ and all possible separations of the vector α on sub vectors (vectors of the form $(\tilde{a}_i, \tilde{a}_{i+1}, ..., \tilde{a}_j)$ when $1 \leq i \leq j \leq n$ To solve the problem, it is possible to use formulas containing the function $sign(x)$. The notation is used: τ_j – is the number of coordinates of the vector $a = (a_1, a_2, ..., a_n)$, large j-th coordinates or equal to it; η_j – is the number of the j-th coordinate of the vector a, after placing the coordinates a in descending order. The relations are used

$$\tau_j = \sum_{i=1}^{n} sign(1 + sign(a_i - a_j)), \quad j = 1,2,...,n \ .$$

Due to the requirement the numbers $\eta_1, \eta_2, ..., \eta_n$ are determined. They can be found according to the formulas

$$\eta_j = 1 + \tau_j - \sum_{i=1}^{n} (1 - (sign(\tau_i - \tau_j))^2 sign(1 + \\ + sign(i - j)), \quad j = 1,2,...,n \ .$$

admit the expression

$$\tilde{a}_j = \sum_{i=1}^{n} a_i (1 - (sign(\eta_i - j))^2), \ j = 1,..,n \ .$$

The desired vector is obtained $\tilde{a} = (\tilde{a}_1, \tilde{a}_2, ..., \tilde{a}_n)$. Next, the relocation $\Theta_1, \Theta_2, ..., \Theta_n$ of the elements of the set $\{1,2,..,n\}$ is carried out as they are corresponding to changing relocation of $n_1, n_2, ..., n_n$ .This relocation is done by taking $\Theta_{\eta_i} = i$ for each $1 \leq i \leq n$. Any partition of the set $\Re$ into m clusters has the form $\{T_j = \{K_{\Theta_{n_{j-1}+1}}, K_{\Theta_{n_{j-1}+2}}, ..., K_{\Theta_{n_j}}\}, 1 \leq j \leq m\}$ which allows to specify it $n_1, n_2, ..., n_{m-1}$ (when $n_0 = 0 < n_1 < n_2 < ... < n_m = n$). The cluster size is T_j the difference is Δ_j the between distance of the largest and the smallest brightness of large pixels are

 ISBN 978-80-86943-42-8

$\Delta_j = \widetilde{a}_{n_{j-1}+1} - \widetilde{a}_{n_j}\ j = 1,2,..,m$. The maximum Δ distances of the clusters will be called the minimal step of the partition: $\Delta = \max_{1 \le j \le m} \Delta_j$.

Fixing an arbitrary non negative number ε. $m(\varepsilon)$ is possible if the smallest number of clusters of the set $\Re$ can be divided under the condition that the smallest dimension of the result partition does not exceed ε. The value $\Delta(\varepsilon)$ is the smallest possible size of partitioning the set $\Re$ into $m(\varepsilon)$ clusters (obviously,it is $\Delta(\varepsilon) \le \varepsilon$). Let us introduce $\sigma(\varepsilon)$ as the smallest possible sum of cluster sizes among all partitions of $\Re$ into $m(\varepsilon)$ clusters with minimum size of $\Delta(\varepsilon)$. This is true, since there exists a partition of the set $\Re$ with the dimension which does not exceed ε. Under the condition that the set of all partitions of $\Re$ is finite.

In this paper, the problem of finding a partition of the set $\Re$ into $m(\varepsilon)$ of minimal clusters $\Delta(\varepsilon)$ with the sum of the ranges of the clusters $\sigma(\varepsilon)$ is solved. The triple minimizing algorithm can be mathematically described and justified. It consists of three stages, each of which minimizes a certain numerical characteristic - the number of clusters, the minimum and the sum of cluster dimensions. At least any of them has an acceptable result. All three minimization are performed for $\le \varepsilon$ and each subsequent minimization preserves the result of the previous minimization.

4. MATHEMATICAL JUSTIFICATION OF THE ALGORITHM

For any non negative number δ, we define δ as the partition of the set $\Re$ as follows. Since the sequences are increasing $\widetilde{a}_1 - \widetilde{a}_i, i = 1,2,...,n.$ thus a unique number is $1 \le n_1 \le n$ so $\widetilde{a}_1 - \widetilde{a}_{n_1} \le \delta < \widetilde{a}_1 - \widetilde{a}_{n_1+1}$ (for convenience will be assumed $\widetilde{a}_{n+1} = -\infty$). The next number n_j is to be found, where $j \ge 1$. If $n_j = n$ the required δ partition $(n_1, n_2, ..., n_{m-1})$ is already constructed; so

$m := j$. In the case of $n_j < n$, it is possible to use n_{j+1} the only number corresponding to the condition $\widetilde{a}_{n_j+1} - \widetilde{a}_{n_{j+i}} \le \delta < \widetilde{a}_{n_j+1} - \widetilde{a}_{n_{j+i}+1}$. After a finite number of steps, it is possible to obtain a δ breakdown of $(n_1, n_2, ..., n_{m-1})$, and the number of its clusters m.

The algorithm for partitioning into local areas can be realized in three steps.

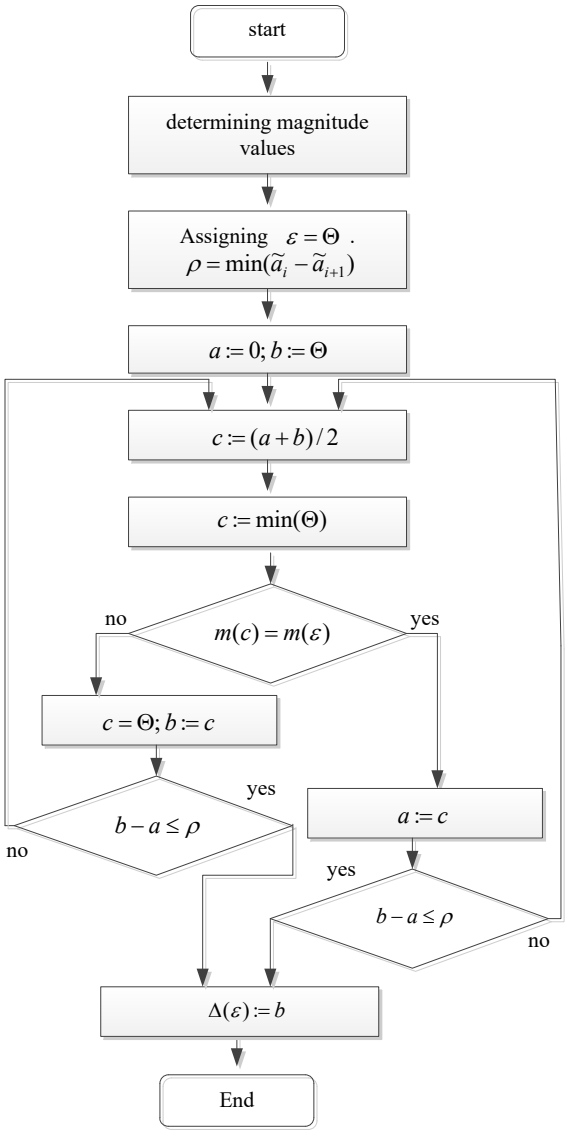


Fig. 1. The algorithm of determining the magnitude of the $\Delta(\varepsilon)$.

I. Calculating the value $m(\varepsilon)$. It is possible to construct an ε-partition of the set $\Re$ and fix the number of its clusters. This value coincides with the value of $m(\varepsilon)$, at the first stage of the algorithm.

II. Calculating the value $\Delta(\varepsilon)$. The algorithm is derived from the construction of the 0-partition, the ε-partition of the set $\Re$, and the calculation of each of the cluster numbers . The values of $m(0)$ and $m(\varepsilon)$ are equal. The minimum of the 0-partition and the size of all its clusters are equal to zero, i.e they assume the values that are minimal for these values. Hence, for $m(0) = m(\varepsilon)$, the equalities $\Delta(\varepsilon) = \sigma(\varepsilon) = 0$, are true, i.e., 0-partition is a finite result of applying the algorithm. For condition of

 ISBN 978-80-86943-42-8

$m(0) \neq m(\varepsilon)$, $\varepsilon > 0$ and $m(0) = m(\varepsilon)$. Having found the minimum of the ε-partition of $\Re$, its value is assigned to the variable Θ.

It is denoted by ρ the smallest of the numbers $(\widetilde{a}_i - \widetilde{a}_{i+1})$, where $1 \le i \le n-1$, $\widetilde{a}_i > \widetilde{a}_{i+1}$ (the inequality $\Theta > 0$ implies the existence of a number of the specified type). Obviously it is $\rho > 0$. The algorithm of determining the magnitude of the $\Delta(\varepsilon)$ is shown in Figure 1.

The algorithm shown in Figure 1 is implemented by means of the following steps:

1) Assigning is $a := 0; b := \Theta$;

2) Assigning is $c := (a+b)/2$;

3) Constructing a *c*-partition and the assignment of the variable Θ to its minimum value;

4) When $m(c) = m(\varepsilon)$, assigning is $c = \Theta; b := c$. If , $m(c) = m(\varepsilon)$ assigning is $a := c$.

5) If the inequality is true then $b - a \le \rho$.

Assigning is $\Delta(\varepsilon) := b$, otherwise it is necessary to return to step 2.

III. The construction of a partition of the set $\Re$ into $m(\varepsilon)$ clusters of minimal dimension $\Delta(\varepsilon)$, with the sum of the cluster sizes $\sigma(\varepsilon)$. Let $m = m(\varepsilon)$ and $\Delta = \Delta(\varepsilon)$ – the values can be found in the first two stages of the algorithm; $(n_1, n_2, ..., n_{m-1})$ Δ-partition of the set $\Re$ into *m* clusters. The third minimization is realized by the recurrence relation method. It is possible to fix arbitrary numbers $1 \le p \le n$, $1 \le M \le m$ and take the notation $\Re_\rho = \{K_{\Theta_1}, K_{\Theta_2}, ..., K_{\Theta_\rho}\}$. The Δ-partition clusters of the set $\Re_\rho$ coincide with the Δ-partition clusters of the set $\Re$. Only the last cluster of the partition $\Re_\rho$ can be part of a cluster partition of $\Re$ with the same number. There is a unique number $1 \le j \le m$ so as $n_{j-1} + 1 \le \rho \le n_j$. $\Re, \delta, m$ can be replaced by $\Re, \Delta, M$ respectively. In this case, it is possible to replace $m(\Delta)$ by *j*. As a result, it is pointed out that it is possible to divide the set $\Re_\rho$ into *M* clusters of size $\le \Delta$ if and only if $j \le M \le \rho$. The equivalence of this condition for double inequality $M \le \rho \le n_M$ is satisfied. The number $1 \le q \le \rho$ is said to be admissible if there exists a decomposition of $\Re_\rho$ into *M* clusters of minimal dimension not exceeding Δ, with *M*-the cluster $\{K_{\Theta_q}, ..., K_{\Theta_\rho}\}$. For *q* to be admissible, it is necessary and sufficient that: 1) the set $\Re_{\rho-1}$ can be partitioned into $M-1)$ clusters $\le \Delta$, and 2) the range Δ_M of the *M* cluster does not exceed Δ. Similarly, it is possible to prove the equivalence of condition (1) to the double inequality of $M - 1 \le q - 1 \le n_{M-1} \Leftrightarrow M \le q \le n_{M-1} + 1$. As *q* increases from *M* to ρ, the quantity of $\Delta_M = \widetilde{a}_q - \widetilde{a}_\rho$ decreases. In the case if $q = \rho$, $\Delta_M = 0 < \Delta$. Consequently, among the values of *q* under consideration there is the smallest *Q* for which the inequality is $\Delta_M \le \Delta$. The condition (2) will correspond only to numbers of $Q \le q \le \rho$. In the case of $\widetilde{a}_M - \widetilde{a}_\rho \le \Delta$, inequality of $\Delta_M \le \Delta$ is satisfied by all values $M \le q \le \rho$ and $Q = M$.

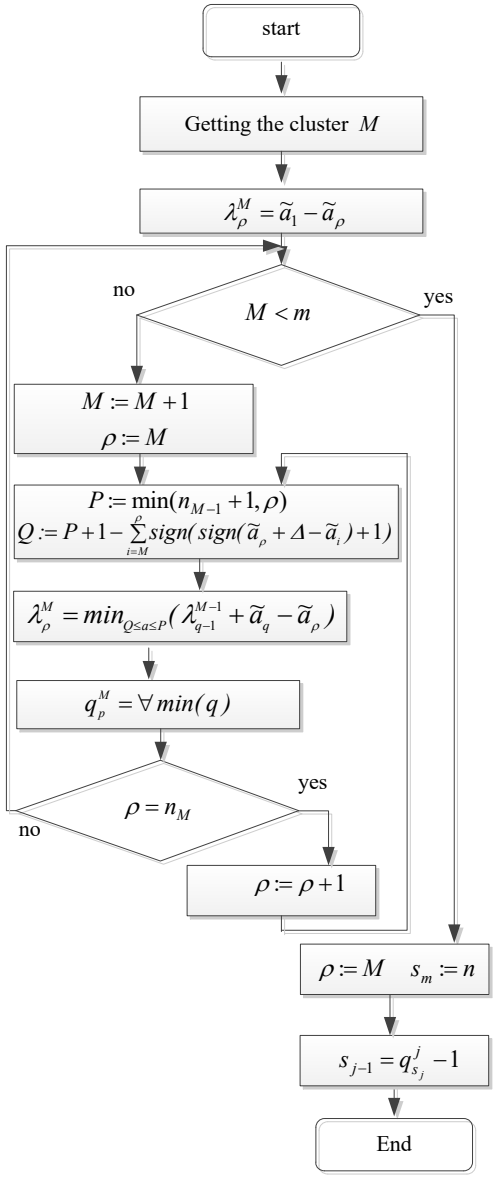


Fig.2. The algorithm of performing step 3.

ISBN 978-80-86943-42-8

If $\widetilde{a}_M - \widetilde{a}_\rho > \Delta$ number $Q(>M)$ is the unique solution of such a system of inequalities with unknown q: $\widetilde{a}_q - \widetilde{a}_\rho \le \Delta$, $\widetilde{a}_{q-1} - \widetilde{a}_\rho > \Delta \Leftrightarrow \widetilde{a}_q - \Delta \le \widetilde{a}_\rho <_{q-1} -\Delta$. Hence the range of allowed values q: $Q \le q \le P$, and $P := \min(n_{M-1}+1, \rho)$. The algorithm for performing step 3 is shown in Figure 2.

The algorithm shown in Figure 2 is implemented in the following steps:

1) Assigning $M := 1$. Calculating quantities $\lambda_\rho^M = \widetilde{a}_1 - \widetilde{a}_\rho$ at $1 \le \rho \le n_1$.

2) In case of $M := m$ go to step 7). At $M < m$ assigning $M := M + 1$.

3) If $M = m$ assigning $\rho := n$. In case of $M < m$ assigning $\rho := M$.

4) Assigning $P := \min(n_{M-1}+1, \rho)$, $Q := P + 1 - \sum_{i=M}^{\rho} sign(sign(\widetilde{a}_\rho + \Delta - \widetilde{a}_i) + 1)$

5) Calculating value $\lambda_\rho^M = \min_{Q \le a \le P}(\lambda_{q-1}^{M-1} + \widetilde{a}_q - \widetilde{a}_\rho)$ and assigning variable q_ρ^M of one (any) of the values of q at which the specified minimum is reached.

6) At $\rho = n_M$ go to step 2). Otherwise – assigning $\rho := \rho + 1$ and go to step 4).

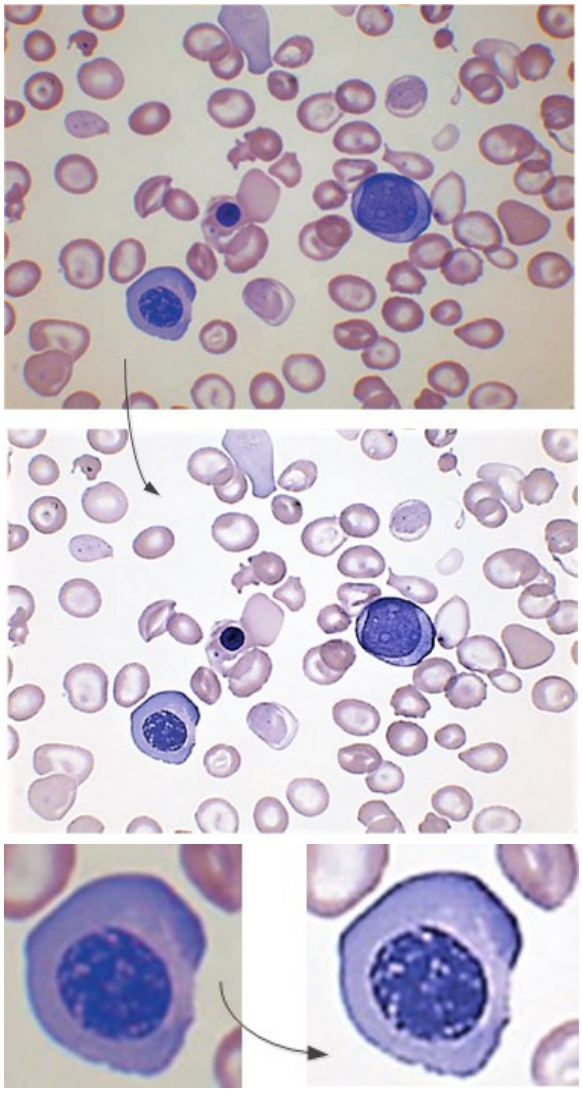
7) Assigning $s_m := n$ and then perform a sequential computation of the numbers $s_{m-1}, s_{m-2}, ..., s_1$ by the recurrence formula $s_{j-1} = q_{s_j}^j - 1$. The resulting partition $(s_1, s_2, ..., s_{m-1})$ is optimal, that is, it has the minimum value of Δ and the sum of the cluster sizes $\sigma(\varepsilon)$.

5. RECOMMENDATIONS FOR THE CHOICE OF ALGORITHM PARAMETERS

Recommendations for choosing the value of ε for calculation. For each of the numbers $\rho = 3$, $\rho = 4$, denoted by $Q_\rho(m)$ the maximum modulus of brightness increment of a large pixel are appeared in the transition from the second image to the ρ–th. We get $Q_3(m) = \max_{1 \le j \le m} \max_{s_{j-1}+1 \le i \le s_j} |\rho_j - a_i| = 0.5\Delta(\varepsilon)$ (due to the optimal partition $(s_1, s_2, ..., s_{m-1})$); $Q_4(m) = \max_{1 \le j \le m} \max_{s_{j-1}+1 \le i \le s_j} |\gamma_i - a_i|$. It is necessary to define the third image with the value $Q_3(m) \le \delta$, it is enough to apply the algorithm for $\varepsilon = 2\delta$. This will ensure a given degree of proximity of the second and third images. Constructing the fourth image, corresponding to the condition of $Q_4(m) \le \delta$ is possible by means of the choice $\varepsilon = \delta$ to guarantee its observance for limiting the brightness of small pixels to 8 bits (range of values 0, 1, 2, ..., 255). The quantities β_j, γ_j will belong to the interval [0, 255], however, they can take fractional values. In this case, they should be rounded according to the formula $x := [x + 0.5]$ ($[t]$ – is the integer part of the number *t*). As a result, the shift of brightness β_j, γ_j to the right or left along the numerical axis can reach 0.5, which requires decreasing of $\varepsilon = 2\delta$ by one, and $\varepsilon = \delta$ by 0.5.

6. EXAMPLES OF PROCESSING

In Figure 3, an example of changing the bit depth of the areas in the image is presented.



ISBN 978-80-86943-42-8

Fig.3. The result of the approach

Asit can be seen from the results presented in Figure 3, this approach allows fulfilling the bit change operation for local domains on images.

7. CONCLUSION

As a conclusion, I want to note the following shortcomings and generalize the advantages. The proposed algorithm allows changing the bit depth in the selected area of the image, however, the select of this section is performed by the operator and is not automated. When processing the entire image, if the threshold is not chosen correctly, artifacts appear in the form of noise. In the continuation of the work, we see the introduction of automation in the process of searching and defining boundaries. It is also possible to modify the algorithm to use an adaptive threshold setting that allows for partitioning into local areas.

8. ACKNOWLEDGMENTS

This work was supported by Russian Ministry of Education and Science in frame of the Federal Program "Research and development on priority directions of scientific-technological complex of Russia Federation in 2014-2020" (contract № 14.577.21.0212 (RFMEFI57716X0212)).

9. REFERENCES

[1] Everitt, J. H., et al. "A three-camera multispectral digital video imaging system." Remote sensing of environment 54.3 (1995): 333-337.

[2] Lloyd, J. Michael. Thermal imaging systems. Springer Science & Business Media, 2013.

[3] Jack, Michael D., Michael Ray, and Richard H. Wyles. "Integrated IR, visible and NIR sensor and methods of fabricating same." U.S. Patent No. 5,808,350. 15 Sep. 1998.

[4] Kanade, Takeo, et al. "A stereo machine for video-rate dense depth mapping and its new applications." Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on. IEEE, 1996.

[5] Chen, Min, Shiwen Mao, and Yunhao Liu. "Big data: A survey." Mobile Networks and Applications 19.2 (2014): 171-209.

[6] Semenishchev, Evgeny A., et al. "Stitching algorithm of the images acquired from different points of fixation." Digital Photography XI. Vol. 9404. International Society for Optics and

ISBN 978-80-86943-42-8

Compression for texture images in different basis functions using system criteria analysis

A.Yu. Babikov
Southern Federal University, Nekrasovski 44, 347922, Taganrog, Russian Federation;
alex12345ander@yandex.ru

V.V. Voronin
Don State Technical University
Gagarin's square 1
344000, Rostov-on-Don, Russian Federation
voroninslava@gmail.com

V.P. Ryzhov
Southern Federal University, Nekrasovski 44, 347922, Taganrog, Russian Federation;
vpr_trtu@mail.ru

Yu.V. Ryzhov
Southern Federal University, Nekrasovski 44, 347922, Taganrog, Russian Federation;
vpr_trtu@mail.ru

ABSTRACT

This paper considers image compression for texture images. For texture representation we consider the orthogonal decomposition of two-dimensional signals (images) using spectral transform in the different basis functions. This paper focuses on the analysis of the following basis DCT, FFT, Haar, Hartley, and Walsh using system criteria analysis. The error of the orthogonal representation of images and the computational cost are considered when choosing a basis system, which is based on the Bellman-Zadeh concept using fuzzy sets. It is shown that the Haar transform can represent textural images more efficiently with smaller average risk than other basis functions.

Keywords

Images, textures, orthogonal bases, fuzzy sets, the Bellman-Zadeh concept

1. INTRODUCTION

Two-dimensional signals (images) are the most important class of signals, since a significant part of the documentation, photographs, illustrations to artistic texts, drawings and much more in human activity refers to this class. This class of two-dimensional signals occupies a significant (sometimes basic) volume of business and household information. Therefore, despite the significant and ever-increasing possibilities of computer technology, in the foreseeable future the requirement of compress the scope of information of such signals exists and will exist.

One of the most effective types of signal compression is spectral compression algorithms based on the use of systems of orthogonal basis functions (from now on abbreviated as bases). This type of compression is widely described in the literature (for example, in [Sof03]). Nevertheless, many problems have not been resolved to the present.

The problem of a well-founded choice of the orthogonal system of functions is not solved yet because of the many possible selection criteria. Earlier it was shown ([Vor16]) that when solving practical problems in conditions of limited time or other resources, it is expedient to use system criteria, including not only the error of signal representation but also the consumption of computational resources, functional stability, and other system-technical characteristics. Such a systematic approach requires consideration of many aspects of the performance and operation of the system (including interaction with other systems, with the environment) and the use of poorly structured expert information. A significant amount of work is devoted to this problem, but they do not contain specific design techniques at the stage of solving signal problems.

The most important task of primary signal processing is the choice of the form of representation [Zha00]. Many applied problems are solved using the time representation, but the relations in the spectral domain build most of the signals processing algorithms [Buj17]. Most systems require the calculation of convolutional integrals that are most efficiently computed by Fast Fourier Transform algorithms. In this case, the choice of the basis of the spectral expansion is essential. Thus, the number of terms of the series is minimized when using the Karhunen–Loeve basis, but the computational speed is significantly increased, and computational costs are reduced when using the bases of piecewise-constant functions (Walsh, Haar). The wide application of the basis of trigonometric functions is due to both the physical and familiarity of such a representation, and the most compact solution of subsequent filtration problems.

ISBN 978-80-86943-42-8
https://doi.org/10.24132/CSRN.2018.2803.11

The formalized solution of the optimal (in the system-technical sense) choice of basis in [Vor16] is largely due to the use of image models in the form of a random two-dimensional Gaussian field with a given correlation function. Such a model is easily implemented on a computer; it can be generated an almost unlimited number of times with the reproduction of the specified parameters, which is convenient for comparing different processing algorithms. But in many cases the nature of transmitted and processed images is different – these are objects of human activity, urban or natural environment, art objects. Textures are often used as an intermediate variant between abstract computer models and real images [Har79], [Pav16]. Also, in the final analysis, the perception of images is based on the features of the human vision as the ultimate recipient and the ultimate decision-making body. Therefore, the task of jointly examining the objective and subjective characteristics of images and the choice of basic systems, considering these two groups of data, is posed in this research.

2. PROPOSED SOLUTION

Let $Z: D \subset R^2 \to R^l$ be a given source image ($l = 1$: for a gray-scale image; $l = 3$: for a RGB color image).

We used 4 different class of textures [Har79] (Fig. 1). They are SKI (clouds), WATER (water), STONE (two types of stones).

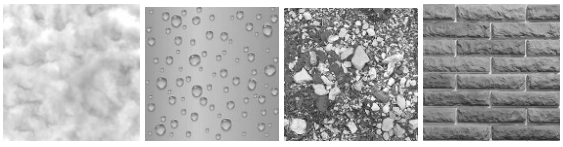


Figure 1. Textures.

For each image, the analysis was performed by the generalized Fourier transform in some basis:

$$G(n,m) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} Z(x,y) \cdot \varphi_{nm}(x,y); \qquad (1)$$

where N, M is the number of counts in coordinates x, y, image $Z(x,y)$, $G(n,m)$- image spectrum in the basis $\varphi_{nm}(x,y)$.

The inverse transform allows you to get the original picture:

$$Z(x,y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} G(n,m) \cdot \varphi_{nm}(x,y). \qquad (2)$$

If the number of employed terms of the series (1) is less than the number of samples (M, N), then the compressed image $\tilde{Z}(x,y)$ differs from the original, calculated by the formula (2). Then the research task was to evaluate the quality of the compressed image.

For each type of texture, the spectrum was calculated in five orthogonal bases: discrete cosine transform (DCT), trigonometric basis (FFT), Haar, Hartley, Walsh. Then the synthesis was performed using the truncated series of images $\tilde{Z}(x,y)$ that were presented to the experts.

3. EXPERIMENTAL RESULTS

Expert assessments

A group of 10 experts was organized to obtain expert assessments. Initially, the original images $Z(x,y)$ (each of the four species), and then synthesized $\tilde{Z}(x,y)$ by truncated series, were presented to the experts. Experts should specify from what number of the removed terms of the series the image can be estimated as excellent, good, satisfactory or unsatisfactory (on a four-point scale).

Nearby the estimate for each basis, the experts had to put the number of terms of the series at which the image remained in the zone of the corresponding estimate. Such criteria as the average value for the number of terms of the series removed and the root-mean-square value of the spread of estimates were determined from the questionnaires of all experts (Table 1). Here, for each image and each basis, the mean values of the bounds of the estimates are given and, through the sign of "/" – the root-mean-square values of the spread of the estimates. The results of calculating the root-mean-square errors associated with the truncation of the series for the textures STONE-1, STONE-2 are shown in Fig. 2, 3.

In many cases of practical importance, it is impossible to obtain all the information from expert practitioners in the form of numbers. Expert opinions can be compactly expressed using the theory of fuzzy sets.

Image / Basis	SKI-28	WATER-2	STONE-1	STONE-2
DCT	53/23 82/19 106/7	51/23 79/13 96/7	139/42 168/37 199/25	125/38 173/59 211/18
FFT	58/16 88/18 102/13	63/16 84/15 100/12	120/41 159/40 194/33	166/31 193/27 216/18
Haar	55/11 74/13 95/10	44/18 72/17 92/12	98/41 142/48 184/37	93/46 153/39 204/28
Hartley	5/3 10/11 83/42	5/2 24/21 67/35	6/3 13/5 77/57	5/3 25/31 128/85
Walsh	37/19 72/9 97/12	51/17 75/14 98/7	80/35 142/46 200/25	92/23 149/33 202/22

Table 1. Results of the expert evaluation of textures

ISBN 978-80-86943-42-8

The main characteristic of fuzzy sets is the membership function of $\mu_A(u)$ an element *u* to the subset A, which assigns to each element of the set functions can be determined by interviewing the experts (system designers) using known scaling techniques.

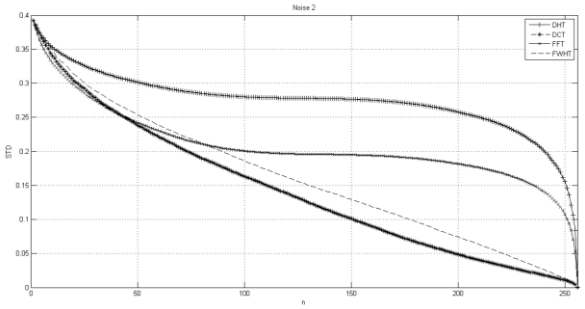
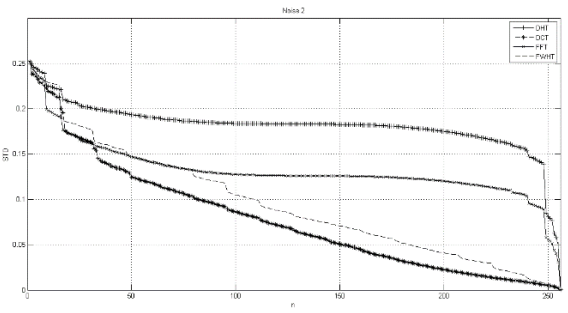


Figure 2. The root-mean-square errors with truncation of the series, Fourier for the STONE-1



texture.

Figure 3. The root-mean-square errors with truncation of the series, Fourier for the STONE-2 texture.

Calculation system

The possibilities of achieving the system's objective that is the implementation of the required system characteristics, and the integration of system-technical constraints are combined in the Bellman-Zadeh concept [Bel76], according to it the fuzzy set $D = C \cap H$ (C is the subset of objectives, H is the subset of constraints) with the membership function $\mu_d(u)$:

$$d_{opt} \in D:\ \mu_d(u) = \mu_c(u) \wedge \mu_h(u) = max \qquad (3)$$

It is called a solution, and the optimal value corresponds to the largest value $\mu_d(u)$. Thus, the objectives and constraints are merged, which allows optimizing the solution quite simply.

When solving the problem of choosing the spectral description of images, it is possible to use the one-dimensional (for square matrices *N*x*N*) or two-dimensional (for *N*x*M* matrices) the space of alternatives, which was used as the number of terms of the truncated series *l*.

$u \in U$ a number from the interval [0, 1] that characterizes the degree of belonging of *u* to the set $A \subset U$. The membership

The membership functions of objectives were defined as the dependence of the image quality on the number of terms used in the Fourier series. It was built by expert assessments. Verbal rates "excellent", "good", "satisfactory", "unsatisfactory" correspond to traditional scores “5”, “4”, “3”, “2”. If you want to increase the significance of any zones of the scale, you can enter non-linear scales or enter the desired non-linear function of the purpose $\mu_c(l)$. In our case, using a semilogarithmic scale of estimates, we could use such an approximation of the data in table 2: $\mu_c(l) = 0{,}5\sqrt{l}$ (curve "T" in Fig. 3). The same figure shows the functions of the purpose $\mu_c(l)$ and constraints $\mu_h(l)$, constructed according to the data of Table. 2 for the SKY-28 image using the FFT basis. In Fig. 4 shows similar membership functions but using the basis of Haar functions.

To introduce the constraint membership function $\mu_h(l)$, we used the number of computational operations (equivalent additions) when computing a given array of spectral components for different bases, these data are given in Table. 2 (from [Vor13]).

The type of basis	**The number of equivalent additions (image size $N \times N$)**
Fourier series transformation	$24N^2 \log_2 N$
Cosine transform	$10N^2 \log_2 N$
Hartley transform	$12N^2 \log_2 N + 5N^2$
Walsh transform	$2N^2 \log_2 N$
Haar transformation	$4N^2 - 2$

Table 2. Number of computational operations of algorithms of two-dimensional spectral transformations

In Table 3 the values of the number of computational operations for bases of trigonometric functions and Haar functions are given as an example.

Because modern computers and microprocessors have achieved high performance, it is possible to smooth out the dependence $\mu_h(l)$ using nonlinear transformations, for example:

 ISBN 978-80-86943-42-8

$$\mu_h(u) = \frac{4}{4 + \lg \frac{S(l)}{S(16)}}, \quad (4)$$

where $S(l)$ is the number of computational operations for l terms of the Fourier series.

Basis/N	16	32	64	128
FFT	$2{,}46 \cdot 10^4$	$1{,}23 \cdot 10^5$	$5{,}88 \cdot 10^5$	$1{,}74 \cdot 10^6$
Haar	$1{,}02 \cdot 10^3$	$4{,}09 \cdot 10^3$	$1{,}64 \cdot 10^4$	$6{,}55 \cdot 10^4$

Table 3. The number of operations in the calculation of spectra in bases of trigonometric functions and Haar functions

The membership of constraints $\mu_h(l)$ using (4) is also shown in Fig. 4 and 5. The abscissa of the intersection point $\mu_c(l)$ and by the Bellman-Zadeh concept allows determining the optimal number of series terms l_{ost} , which is indicated in the figures.

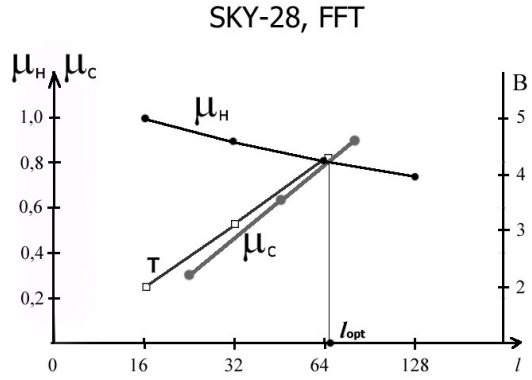


Figure 4. The membership functions of the objective $\mu_c(l)$ and constraints $\mu_h(l)$ for the basis of trigonometric functions.

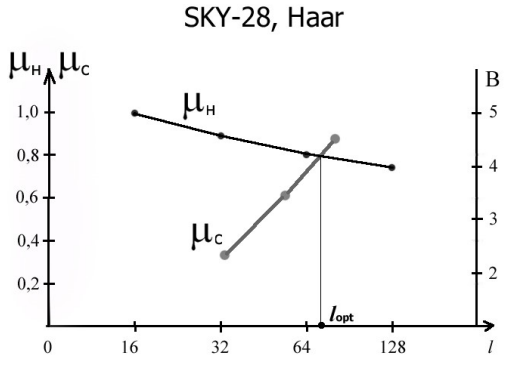


Figure 5. The membership functions of the objective $\mu_c(l)$ and constraints $\mu_h(l)$ for the basis of Haar functions.

4. CONCLUSION

Although the optimal values of the number of series terms turned out to be very close for bases of trigonometric functions and Haar functions, in the Haar basis the number of calculations is less by one or two orders of magnitude, which determines its advantage in using signal detection and recognition problems. However, for tasks related to signal filtering, it is preferable to use the basis of trigonometric functions.

In general, although the expert estimates are correlated with the mean-square error, more acceptable for practice, especially when evaluating objects familiar to humans. When choosing the best basic system, we should consider both the error in the orthogonal representation of images and the laboriousness of the calculations, which is quite simply carried out by the Bellman-Zadeh concept using fuzzy sets.

5. ACKNOWLEDGMENTS

This work was supported by Russian Ministry of Education and Science in accordance to the Government Decree № 218 from April 9, 2010 (project number № 074-11-2018-013 from May 31, 2018 (03.G25.31.0284)).

6. REFERENCES

[Bel70] Bellman R.E., Zadeh L.A. Decision-Making in Fuzzy Environment // Management Science, vol. 17, pp.141- 160, 1970.

[Buj17] Bujack R., Flusser J. Flexible Moment Invariant Bases for 2D Scalar and Vector Fields. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG, pp. 11-20, 2017.

[Har79] Haralick R. Statistical and structural approaches to texture. Proceedings of the IEEE, 67, pp. 786–804, 1979.

[Pav16] Pavie N., Gilet G., Dischler J.-M., Ghazanfarpour D. Procedural Texture Synthesis by Locally Controlled SpotNoise. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG, pp. 71-80, 2016.

[Sof03] Soifer V.A. Methods of computer image processing. Fizmatlit Publ, pp. 784, 2003.

[Vor16] Voronin V.V., Ryzhov V.P., Marchuk V.I., Makov S.V. Texture representations in different basis functions for image synthesis using system criteria analysis. IS&T D International Symposium on Electronic Imaging, Image Processing: Algorithms and Systems XIV 2016, pp. 1-6(6), 2016.

[Zha00] Zhang J., Fieguth P., and Wang D. Random field models. In A. Bovik, editor, Handbook of Image and Video Processing, Academic Press, San Diego, pp. 301–312, 2000.

 ISBN 978-80-86943-42-8