Peter Sykora - Patrik Kamencay - Robert Hudec - Miroslav Benco *

A NEW ALGORITHM FOR KEY FRAME EXTRACTION BASED ON DEPTH MAP USING KINECT

In this paper, a new algorithm for key frame extraction based on depth map for hand gesture recognition is presented. The all input sequences are captured by Microsoft Kinect camera system. These methods extract three key frames from captured depth video sequence. These key frames describe dynamic gesture. The proposed extraction method is composed of two parts. The first part, labelled as space segmentation extracts the region of hand from background. The second part labelled as time segmentation splits captured sequence into three parts and marks one frame per part as the key frame. A new gesture database for evaluation of proposed method was created. The proposed method to human evaluators was compared. The experimental results show that the proposed system obtained accuracy about 90%.

Keywords: *Gesture recognition, 3D video, Microsoft Kinect, depth map, image segmentation.*

1. Introduction

The hand gesture recognition, as a part of non-verbal communication methods of humans, can be useful in human-machine interaction. In many situations, the voice command cannot be executed, e.g. a noisy environment, a person cannot use voice, etc. Methods described in this paper aim for the pre-processing part of the gesture recognition. There are many steps for gesture recognition calculations. First step is the recording and pre-processing. These methods take raw 3D video from Microsoft Kinect and extract only frames that are relevant for dynamic gesture description. The rest of the paper is organised as follows: Section 2 gives a brief overview of the state-of-the-art in gesture recognition problem. The proposed methods are described in section 3. Finally, experimental results, implementation issues and conclusions are discussed in Section 4, or in Section 5.

2. Related Work

A lot of present gesture recognition systems use some hardware peripheries to estimate the gesture variables, e.g. data-gloves [1 - 2]. Other approaches use markers to track the position of various parts of hand, mostly the fingers [3 - 5]. Even the systems working with video data use some gloves or markers to track the position [6]. There are some difficulties in segmentation of hand. For example, the skin of two humans can vary in dominant colour [7]. There are some other problems related to the hand description following feature extractions and their classification. Some methods relay on physiology of human hand [8]. Likewise, the gesture can be represented by motion of hand [9]. The aim of our research is to improve the methods for gesture feature extraction.

3. Proposed System

In this section the proposed system for key frame extraction is presented. The flowchart of the proposed system is illustrated in Fig. 1. The space segmentation (blue section) is the first part of system. The hand region is extracted from video sequence. Subsequently, hand sequences are processed by time segmentation part of the system (green section). This part splits the sequence to three sections. Next, the extraction of a defined number of key-frames is provided. It is assumed that these frames represent the entire sequence. These frames are used in another part of the overall hand recognition system.

* **Peter Sykora, Patrik Kamencay, Robert Hudec, Miroslav Benco**
Department of Telecommunications and Multimedia, Faculty of Electrical Engineering, University of Zilina, Slovakia
E-mail: sykora@fel.uniza.sk

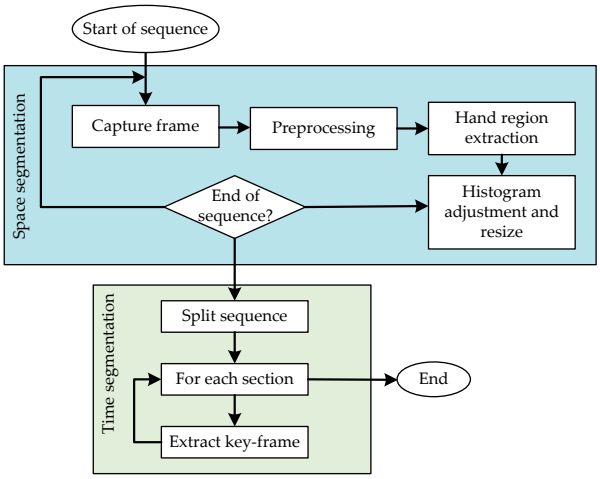

Fig. 1 Flowchart of the segmentation algorithm

The scene is captured by the camera as a first step. It is a 640 by 480 resolution depth map picture with depth of 16 bits. Pixels with the zero value cover locations of the scene when Kinect camera fails to calculate the distance. As zero values represent errors, high values represent a region with high distance from camera. Thus the hand region falls between these two extremes. The values of nonzero pixels are divided from the maximum of 16 bit (65535). Now the image appears as the disparity map. The low values of pixels represent background and errors of depth map. Next, the histogram is stretched to highlight details (Fig. 2).

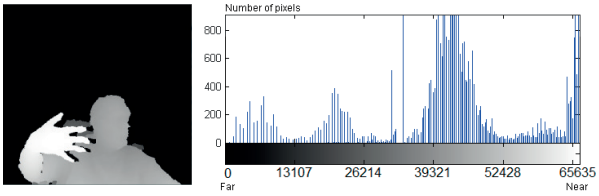

Fig. 2 Disparity map of captured scene and its histogram

3.1 Space Segmentation

The finding of the nearest pixel to the camera is the first step of the space segmentation. The value of pixel is defined as the distance of represented object from the camera. The algorithm finds pixel with the highest value. This value is stored as maxim. From this maxim the border is calculated as border=maxim-3500. The value 3500 is set subjectively and it represents the depth of segmented object. For the human hand a depth of all pixels with a value lower than this border is set to zero. The image now contains only the hand and black background. After this, pixel values of the hand are in the close range of high values. Stretching the histogram of this image is the next step. The details of third dimension are more visible after this. Cropping the excessive areas is the next step. The resizing of the image is the last part. To avoid the hand to be stretched, additional black pixels are added. Resulted image can be seen in Fig. 3.

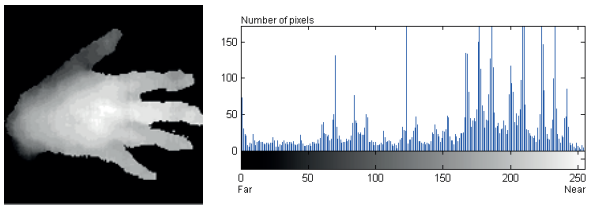

Fig. 3 Resulted picture of one frame and its histogram

3.2 Time Segmentation Methods

When the whole duration of dynamic gesture is captured and segmented, the key frame extraction part is started. These algorithms pick up a defined number of images from the sequence. Frames are picked up by their statistical position in the sequence (described below). Firstly, the methods split the sequence to more parts based on the significant change of shape. For this system, all methods divide the sequence to three parts (begin, centre and end as seen in Fig. 4).

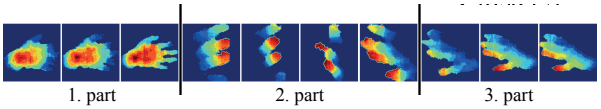

Fig. 4 Depth map sequence splitting to parts

A method, labelled as D_A, calculates Euclidean distance of neighbour frames. This is a distance of two vectors (frames) in high-dimensional space. If sequence has 14 frames, there are 13 neighbour distances. Let $\boldsymbol{d}$ be a vector of distances, **S** be a sequence of frames of length N and *mean* be the function that calculates mean number of a vector. Thus $\boldsymbol{d}_i$ is calculated as seen in following formula:

$$d_i = mean(S_{i-1} - S_i), i \in \{0, 1, ..., N-1\}. \quad (1)$$

A method, labelled M_A, calculates division image of two neighbouring frames. Next it calculates the mean pixel value of this image. Let $\boldsymbol{m}$ be a vector of distances and L_2 be the function that calculates Euclidean distance of two vectors. Thus mi is calculated as seen in Equation 2.

$$m_i = L_2(S_i, S_{i+1}), i \in \{0, 1..., N-1\}. \quad (2)$$

When visualised as curve, the highest peak determinates the first division of sequence. It can be seen as top green circle in Fig. 5. After remembering its position, this value is

set to zero. New highest peak determinates second division (purple circle). The highest peaks in first or last frame can be problematic. There are no frames before first frame. To resolve this, algorithm ignores the first and last values from curve (blue circles). Also, after zeroing first peak, neighbour values are set to zero too (red circles). At least one frame in the cut is ensured by this.

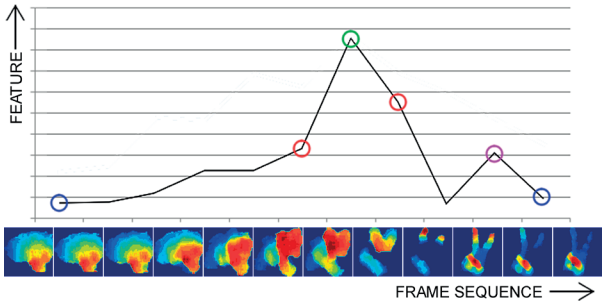

Fig. 5 Visualisation of significant points in the distance vector

Secondly, the key frames are extracted from the cuts. Two methods are used again. First method is labelled D_B. It calculates Euclidean distance for all combinations of frames in the sub-sequence. Frame with the lowest values to all other frames is picked up. It is the best representation for this sub-sequence. Let **Q** be the section of sequence with length *M*. The resulted key-frame **Fk** is picked as the frame with the smallest value in overall distance vector ***a*** (Equation 3). The vector ***a*** is calculated as sum of columns from distance matrix **A** (Equation 4). The position in matrix **A** represents the Euclidean distance of two frames of sequence. E.g. A at 1, 2 positions represent distance between frames 1 and 2:

$$\mathrm{F}_k = \mathrm{S}_i, i \rightarrow a_i = \min(a), \tag{3}$$

$$a_i = \sum_{k=0}^{M-1} \mathrm{A}_{ki}, i,k \in \{0,1,...,M-1\}, \tag{4}$$

$$\mathrm{A}_{ij} = \begin{cases} L_2(\mathrm{Q}_i, \mathrm{Q}_j), i \neq j \\ 0, \quad otherwise \end{cases}; i,k \in \{0,1,...,M-1\}. \tag{5}$$

The second method is labelled M_B. It calculates the mean pixel value from all sub-sequence pixels. This creates the so called mean picture. The mean picture is composed of mean pixels. The frame from sequence is taken. Pixel from this frame is compared with pixel from the mean picture. The equal counter is increased if they are the same. Each frame has this counter. The highest counter points to the frame that has the most pixels as the mean picture. This method ignores the zero valued pixels as they represent the background. Key-frame **Fk** as frame with maximal value in counter vector ***b*** is picked (Equation 6). The value in vector ***b*** on position *i* is incremented if pixel value of frame **P** is the same as in **MED** picture (Equation 7). As the calculation goes pixel by pixel, the vector **P** stores pixel values at given position from whole section **Q** (Equation 8). The **MED** picture is calculated by median value of pixels of section **Q** at given position:

$$\mathbf{Fk} = \mathrm{S}_i, i \rightarrow b_i = \max(\boldsymbol{b}), \tag{6}$$

$$b_i = \begin{cases} b_i = b_i + 1, \mathbf{MED} = \mathrm{P}_i \\ b_i \qquad otherwise \end{cases}; i \in \{0,1,...,M-1\}, \tag{7}$$

$$\mathrm{P}_i = \mathrm{Q}_{xyi} \quad \begin{matrix} x \in \{0,1,...,width\} \\ y \in \{0,1,...,height\} \end{matrix}, \tag{8}$$

$$\mathrm{MED}_{xy} = mean(\mathrm{Q}_{xyi}). \tag{9}$$

From these methods four combinations are created. These algorithms are labelled M_A-M_B, M_A-D_B, D_A-M_B and D_A-D_B.

4. Evaluation of Proposed Methods

The description of experiments and the results of proposed methods are presented in this section. The algorithms are implemented in C++ programming language with OpenCV library in Microsoft Visual Studio. The KinectSDK only for communication with Microsoft Kinect camera system is used. The experiment was run on the Microsoft Windows 7 Professional 64-bit computer with two AMD Opteron 6134 processors and 16 GB RAM memory.

4.1 Database of Depth Video Sequences

To compare these methods the database of depth video sequences is created. The existing gesture databases contain static gestures, or the dynamics is in the motion of hand rather than in the shape change [10 - 12]. Ten dynamic gestures were presented by 10 actors for this database. From them 3 were females and 7 were males. Total 100 depth video sequences are in dataset. The Microsoft Kinect camera was used for capturing. The sequences don't have the same length but they are processed by the first stage of segmentation. They have the same resolution of 150 by 150 pixels. Captured sequences of dynamic hand gestures are shown in Fig. 6. For this example, all have the same length. For example, the first gesture (first row in Fig. 6) is the hand waving from down to up. The second gesture (second row in Fig. 6) is in reverse. More gestures have the same shape change but in different direction. Gestures of such pairs are in rows 1-2, 3-4 and 9-10.

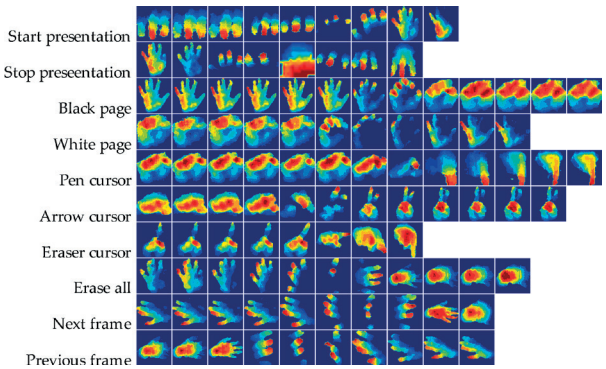

Fig. 6 Short example sequences of all gestures

4.2 Experimental Results

The goal of the experiment is to segment (extract key frames) all sequences with 4 presented methods. In the result there are 3 images that represent the whole sequence. They contain the information about dynamic shape change of hand. To evaluate these results, 10 respondents were appointed. Each respondent is supposed to divide the sequence to three parts. Next, they pick up the most representative picture of part. If they can't distinguish between multiple images, several similar images can be pointed. This gives 10 reference results per sequence. If system picks up the same picture as some of human evaluators, this result is considered to be correct. It basically means that this system acts as human.

Overall results for the four methods are in Table 1. The best results are represented by bold font. It is clear that method D_A-D_B gives the best results. Figure 7 shows results of all methods by the order of frames. For the first extracted frame all methods give almost the same results as humans. High accuracy is given on the last frames too. The results for the middle frame are worse than for the borders, because the dynamics of shape change is higher. The D_A-D_B method achieves overall accuracy of 90 %.

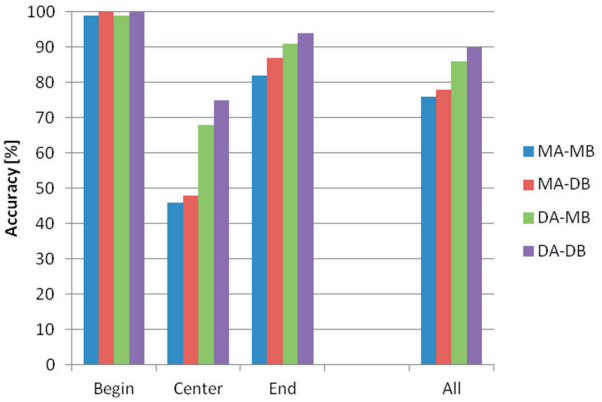

Fig. 7 Graph representing accuracy of time segmentation methods

Next Table 2 shows the time needed for calculation of each method based on the length of the sequence. For testing the length of sequence it is in the range from 8 frames to 30 frames. Results are in milliseconds. Again the method D_A-D_B provides the best conclusion as it is the fastest. For example, method D_A-D_B needs 2.3 ms to calculate three key-frames from sequence long 22 frames. On the other hand, the method M_A-M_B needs 10.3 ms to do the same.

In Fig. 8 these values are shown as graphs. Each method appears to have a trend. The processing time is in direct correlation with length of the sequence. From all these results the method D_A-D_B gives the best accuracy and the fastest calculation.

Accuracy of methods by actor — Table 1

	Actor	0	1	2	3	4	5	6	7	8	9
Accuracy	M_A-M_B	0.77	0.63	0.77	0.87	0.70	0.83	0.73	0.80	0.83	0.63
	M_A-D_B	0.77	0.73	0.83	0.87	0.70	0.87	0.70	0.87	0.83	0.67
	D_A-M_B	0.77	**0.97**	**0.90**	0.97	0.90	0.83	0.80	0.90	**0.90**	0.67
	D_A-D_B	**0.87**	**0.97**	**0.90**	**1.00**	**0.93**	**0.90**	**0.83**	**0.93**	0.90	0.73

Calculation time for each method by the length of the sequence — Table 2

Number of frames		8	12	16	20	24	28	30
Calculation time [ms]	D_A-D_B	**1.400**	**1.516**	**1.759**	**2.386**	**2.198**	**2.612**	**2.992**
	D_A-M_B	7.267	7.509	8.392	8.189	8.636	9.079	8.869
	M_A-D_B	1.572	2.448	2.822	2.890	3.793	3.969	4.299
	M_A-M_B	7.937	8.633	9.770	9.863	10.176	10.676	9.889

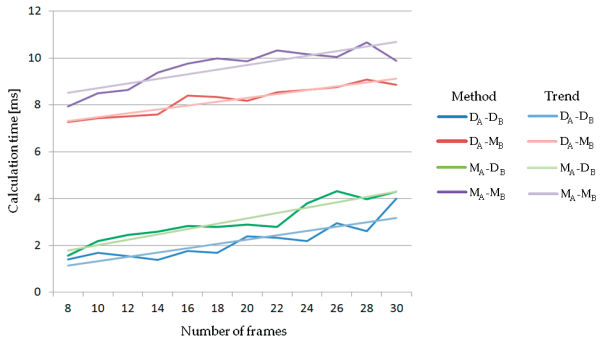

Fig. 8 Graph representing time needed for calculation of each method

5. Conclusion

In this paper a novel segmentation method for hand gesture recognition was described. This method takes raw depth video from Microsoft Kinect camera. It extracts the key-frames for further processing. These frames contain information about the whole dynamic gesture. This system is divided into two parts, the space segmentation and time segmentation. Firstly, the space segmentation extracts the region of human hand from depth video sequence. Next, it resizes the image to have resolution 150 by 150 pixels and stretches the histogram to highlight details. Secondly, the time segmentation extracts the key-frames. From the tests it is clear that system D_A-D_B extracts the frames with accuracy of 90%. As this pre-processing method provides three pictures it is up to the next stage to process them and recognize the whole dynamic gesture.

To the future, we plan to include this algorithm as the first step in dynamic gesture recognition system. The features should be extracted [13] from these three frames and combined for the classification [14].

Acknowledgments

This contribution is the result of the project's implementation at the Centre of Excellence for Systems and Services of Intelligent Transport, ITMS 26220120028. It was supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] YOSHIMURA, Y., OZAWA, R.: *A Supervisory Control System for a Multi-fingered Robotic Hand using Datagloves and a Haptic Device.* Intern. Conference Intelligent Robots and Systems (IROS), 2012, p. 5414-5419

[2] CARBONARO, N., MURA, G. D., LORUSSI, F., PARADISO, R., DE ROSSI, D., TOGNETTI, A.: Exploiting Wearable Goniometer Technology for Motion Sensing Gloves. *IEEE J. of Biomedical and Health Informatics*, 2014, vol. 18, No. 6, p. 1788-1795

[3] MINNEN, D., ZAFRULLA, Z.: *Towards Robust Cross-user Hand Tracking and Shape Recognition.* IEEE Intern. Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1235-1241

[4] CHANG, L. Y., POLLARD, N. S., MITCHELL, T. M., XING, E. P.: *Feature Selection for Grasp Recognition from Optical Markers.* Intern. Conference on Intelligent Robots and Systems, 2007, pp. 2944-2950

[5] TANGSUKSANT, W., ADHAN, S., PINTAVIROOJ, C.: *American Sign Language Recognition by using 3D Geometric Invariant Feature and ANN Classification.* 7th Intern. Conference Biomedical Engineering (BMEiCON), 2014, pp. 1-5

[6] MEYER, J., KUDERER, M., MULLER, J., BURGARD, W.: *Online Marker Labeling for Fully Automatic Skeleton Tracking in Optical Motion Capture.* IEEE Intern. Conference on Robotics and Automation (ICRA), 2014, pp. 5652-5657,

[7] LUKAC, P., HUDEC, R., BENCO, M., KAMENCAY, P., DUBCOVA, Z., ZACHARIASOVA, M.: *Simple Comparison of Image Segmentation Algorithms Based on Evaluation Criterion.* 21st Intern. Conference Radioelektronika (RADIOELEKTRONIKA), 2011, pp. 1-4

[8] QURAISHI, M. I., DHAL, K. G., CHOUDHURY, J. P., GHOSH, P., SAI, P., DE, M.: *A Novel Human Hand Finger Gesture Recognition using Machine Learning.* 2nd IEEE Intern. Conference on Parallel Distributed and Grid Computing (PDGC), 2012 , pp. 882-887

[9] JUAN, W., GUIFANG, Q., JUN, Z., YING, Z., GUANGMING, S.: Hand Motion-Based Remote Control Interface with Vibrotactile Feedback for Home Robots. *Int. J. Adv Robot Syst*, 2013, vol. 10:270. doi: 10.5772/56617

[10] ZHOU, R., JUNSONG, Y., WENYU, L., ZHENGYOU, Z.: Minimum Near-Convex Shape Decomposition and its Application in Hand Gesture Recognition. *Intern. J. of Computer Vision* (*IJCV*), 2011

[11] KIM, T. K., WONG, S. F., CIPOLLA, R.: *Tensor Canonical Correlation Analysis for Action Classification.* Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007

[12] LIU, L., SHAO, L.: *Learning Discriminative Representations from RGB-D Video Data.* Proc. of Intern. Joint Conference on Artificial Intelligence (IJCAI), Beijing, 2013

[13] CHENYANG, Z., XIAODONG, Y., YINGLI, T.: Histogram of 3D Facets: *A characteristic Descriptor for Hand Gesture Recognition.* 10th IEEE Intern. Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013, pp. 1-8

[14] DARDAS, N. H., GEORGANAS, N. D. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *IEEE Transactions on Instrumentation and Measurement*, 2011, vol. 60, No. 11, p. 3592-3607.