Towards Automatic Recognition of Sign Language Gestures Using Kinect 2.0

Dmitry Ryumin[1,2] and Alexey A. Karpov[1,2(✉)]

[1] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, SPIIRAS, St. Petersburg, Russian Federation
dl_03.03.1991@mail.ru, karpov@iias.spb.su
[2] ITMO University, St. Petersburg, Russian Federation
http://hci.nw.ru

Abstract. We present a prototype of a new computer system aimed at recognition of manual gestures using Kinect 2.0 for Windows. This sensor allows getting a stream of optical images having FullHD resolution with 30 frames per second (fps) and a depth map of the scene. At present, our system is able to recognize continuous fingerspelling gestures and sequences of digits in Russian and Kazakh sign languages (SL). Our gesture vocabulary contains 52 fingerspelling gestures. We have collected a visual database of SL gestures, which consists of Kinect-based recordings of 2 persons (a man and a woman) demonstrating manual gestures. 5 samples of each gesture were applied for training models and the rest data were used for tuning and testing the developed recognition system. Model of each gesture is presented as a vector of informative visual features, calculated for the hand palm and all fingers. Feature vectors are extracted from both training and test samples of gestures, then comparison of reference patterns (models) and sequences of test vectors is made using the Euclidian distance. Sequences of vectors are compared using the dynamic time warping method (dynamic programming) and a reference pattern with a minimal distance is selected as a recognition result. According to our experiments in the signer-dependent mode with 2 demonstrators from the visual database, the average accuracy of gesture recognition is 87% for 52 manual signs.

Keywords: Sign language · Assistive technology · Automatic gesture recognition · Image processing · Kinect sensor

1 Introduction

Sign languages (SLs) are known as a natural means for verbal communication of the deaf and hard-of-hearing people. All the SLs use visual-kinetic clues for human-to-human communication combining manual gestures with articulation of lips, facial expressions and mimics. At present there is no universal SL all over the world, and almost each country has its own national SL. Any SL has a specific and simplified grammar, which is quite different from that of spoken languages. In addition to conversational SLs, there are also fingerspelling alphabets, which are used to spell whole

M. Antona and C. Stephanidis (Eds.): UAHCI 2017, Part II, LNCS 10278, pp. 89–101, 2017.
DOI: 10.1007/978-3-319-58703-5_7

words (such as names, rare words, unknown signs, etc.) letter-by-letter. All the fingerspelling systems depend on national alphabets and there are both one-handed fingerspelling alphabets (in France, USA, Russia, Kazakhstan, etc.) and two-handed ones (in Czechia, UK, Turkey, etc.). Russian SL (RSL) is a native communicative language of the deaf people in the Russian Federation, Belarus, Kazakhstan, Ukraine, Moldova, also partly in Bulgaria, Latvia, Estonia, Lithuania, etc.; almost 200 thousand people use it daily. Many of these countries also have own national SLs like Kazakh SL (KSL), which is very similar to RSL. In RSL, there are 33 letters, which are demonstrated as static or dynamic finger signs, and in KSL, there are 9 additional letters (42 letters in total); so the whole gesture vocabulary contains 52 different items.

Thus, the creation of computer systems for automatic processing of SLs such as gesture recognition, text-to-SL synthesis, machine translation, learning systems and so on is a current and useful topic for research. There are quite many recent articles both on recognition [1–3] and synthesis [4, 5] of hand gestures of a sign language and fingerspelling (finger signs) [6, 7]. SL recognition is also one of key topics at the recent HCI conferences [8–11].

At present, Microsoft Kinect sensors are very popular in edutainment domain and they are quite efficient for the task of SL recognition too [3, 12, 13]. MS Kinect 2.0, which is the last version of the video sensor released by Microsoft in 2014, provides simultaneous detection and automatic tracking of up to 6 people at the distance of 1.2–3.5 m from the sensor. In the software, a virtual model of human's body is presented as a 3D skeleton of 25 points.

2 Architecture of the Recognition System

The recognition system must identify a shown gesture as precisely as possible and minimize the recognition error. Figure 1 shows a functional scheme of the system for sign language automatic recognition using MS Kinect 2.0. MS Kinect 2.0 is connected to the workstation via USB 3.0. The viewing angles are 43.5° vertically and 57° horizontally. Resolution of the video stream is 1920 × 1080 pixels with a frequency of 30 Hz (15 Hz in low light conditions) [14]. The inclination angle adjuster is pointed at changing vertical viewing angle within the range of ± 27°. Workstation (a high-performance personal computer) has software GestureRecognition of the sign language recognition system and a database, part of which is used for training, and the rest for experimental studies.

Practical studies revealed that the most optimal model of the database design, capable of storing the multidimensional signals (received from the sensor Kinect 2.0), is a hierarchical model [15]. The root directory of the database consists of 52 subdirectories, 10 of which contain information about gestures showing the numbers 1–10. The rest of the directories store static and dynamic gestures of finger alphabets (Russian and Kazakh dactylology), consisting of 42 letters. The RSL has 33 letters that are shown in the form of static gestures. In the KSL the Russian finger alphabet is supplemented by 9 letters that are reproduced dynamically.

A single subdirectory includes 30–60 video files with one and the same recorded gesture shown by demonstrators many times in a monochrome light or dark-green

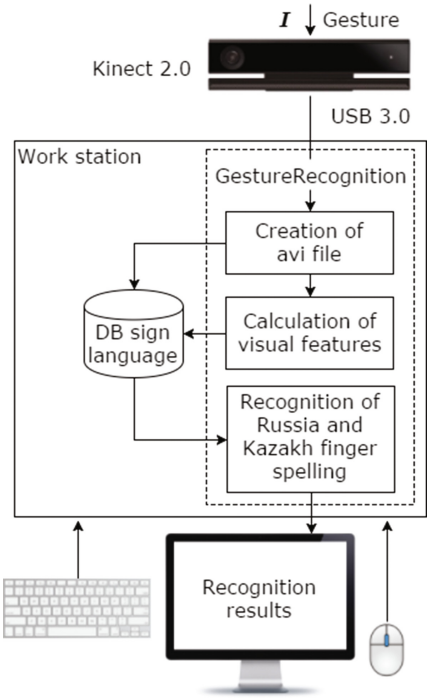

Fig. 1. Functional scheme of the automatic recognition system for sign language

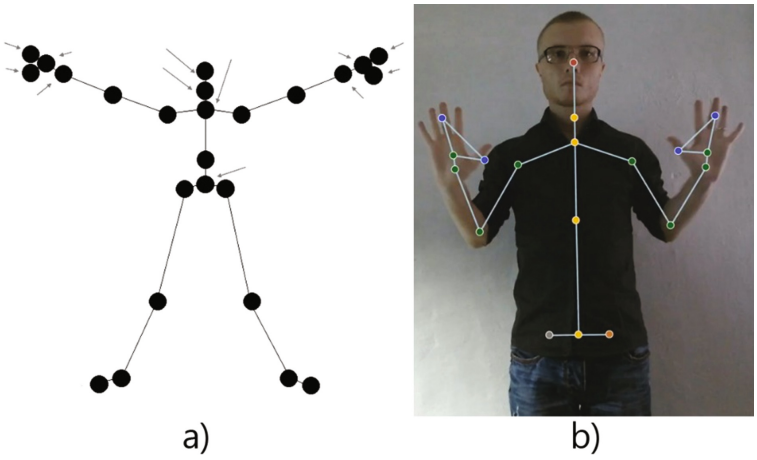

Fig. 2. Human's body model: (a) 25-point model of the human skeleton; (b) detection of a user in the frame

background; the same number of text files with the coordinates of the skeleton pattern (divided into 25 joints) of the detected person. Each definite point is the intersection of the two axes (X, Y) on the coordinate plane (Fig. 2) and the additional value of the Z coordinate with double precision, indicating the depth of the point, which is measured by the distance from the sensor to the object point. The optimal distance is 1.5–2 m. Besides the above-mentioned files, there is also a text file storing service information about the gesture. The average duration of one video file is ≈4–5 s.

The database contains two demonstrators (man and woman), each of whom showed the same gesture 30 times. Examples of video frames depicting demonstrators from the database are shown in Fig. 3. Recognition system training was carried out on the extracted visual signs from 5 video files. These files are considered benchmark gestures, while the rest are used as test data.

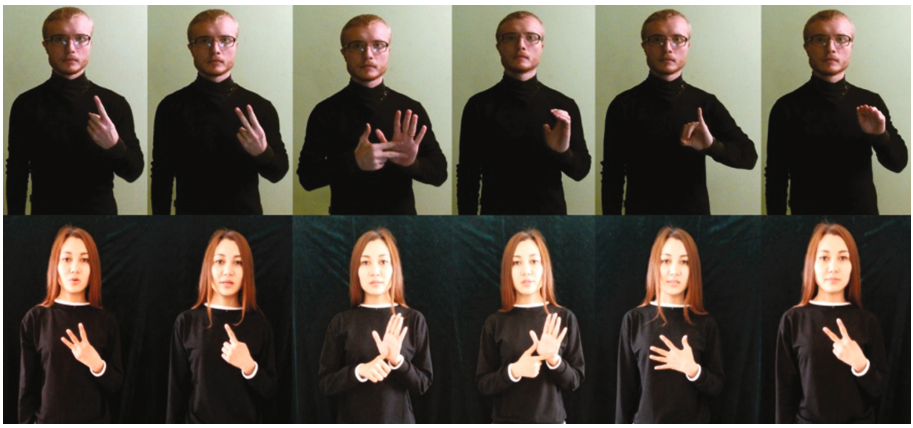

Fig. 3. Examples of video frames depicting two demonstrators from the database

Method for extracting visual cues is as follows. Gesture is delivered to the recognition system as a video signal. Then, for the entire video stream the optimal threshold value of reducing brightness is selected. After this the filling of the inner regions of the objects (closing) occurs. It is based on the following steps:

(1) Searching for one color square that is completely surrounded by the other one;
(2) Replacing the found region with the color of the surrounding region.

This operation is necessary for obtaining solid objects and is given according:

$$A \bullet B = (A \oplus B) \ominus B$$

As can be seen from the expression, first dilation is applied, and the obtained result is subjected to erosion [16].

Removal of minor noise and uneven borders around objects is carried out with the help of masking in the form of a structural element consisting of a matrix of zeros and

ones, forming an oval square. The diameter is selected based on the best results. With increasing the diameter, small objects disappear. Objects, which have color characteristics matching hands color, but much smaller in diameter, are excluded likewise.

Finally, an square, which contains the coordinates from a set of text data obtained from MS Kinect 2.0, will be a hand.

When testing on prerecorded gesture database, it was revealed that the deviations from the normal functioning occur when the hand tilt angle exceeds 45°. This is due to the fact that the sensor MS Kinect 2.0 is unable to identify the key point around the hand center. This problem is solved by the method of averaging the last 7 previous horizontal and vertical peaks of the hand contour, which allows us to predict in what place the peaks will be in the subsequent time moment.

Next, using the skeleton data, obtained from the sensor MS Kinect 2.0, a demonstrator's hand is represented as an ellipse (Fig. 4), on condition that it constitutes one object. The semi-major axis runs through the points that are in wrist, the center of the hand and vertices of middle and ring fingers. The semi-minor axis runs perpendicular to the major axis through the point with the coordinates of the hand center.

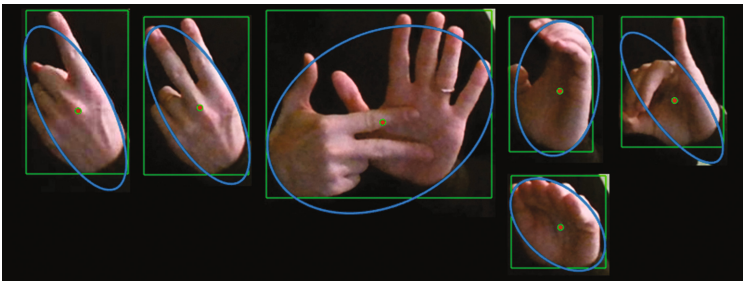

Fig. 4. Representation of hands in the form of an ellipse

After finding the ellipse, the first geometrical informative feature in the form of the orientation of hands is defined. The value is the angle between the x-axis of the coordinate plane and the major axis of the object (hand), as illustrated in Fig. 5.

Other interrelated necessary features are the lengths of the ellipse's axes. It should be noted that MS Kinect 2.0 does not allow for high-precision determination of the coordinates of the vertices, which can be reflected in false values. Therefore, a pre-colored image of hands is transferred into binary. This allows us to separate background from the hand and get the optimum values of lengths of both minor and major axes of the ellipse.

Then eccentricity is determined by dividing the major axis length by the minor axis length. The resulting value is added to the found features.

Other features are the topological properties of the binary object, such as the number of holes inside the object and the Euler number, calculated based on the difference in the number of objects and their inner holes in the image [17].

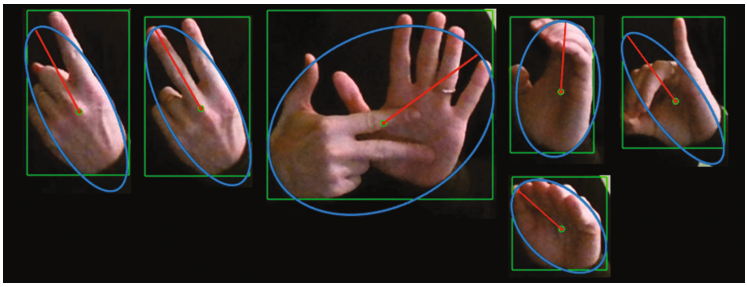

Fig. 5. Orientation of hands (the angle between the x-axis and the major axis of the object)

In continuation of analysis of the binary object, the square and convexity coefficient are determined. The convexity coefficient is defined as the ratio of the square of the object to the square of the quadrilateral that completely encompasses the object.

Using the Sobel operator [18] we obtain hand border (examples are shown in Fig. 6) and find its length and diameter. Finally, we complement an array of features with the values obtained.

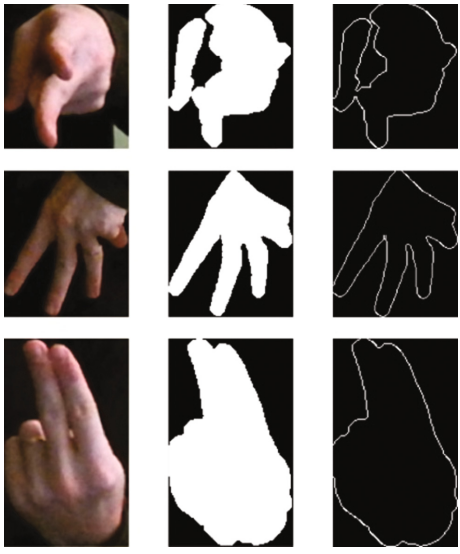

Fig. 6. Examples of determining the hand contour

In total, the defined informative features characterize the hand in the general view, without determination of a shown gesture with a high probability. Therefore, the hand opening process is performed using the structural element in the form of a circle with a diameter of 1/5 of the length of the ellipse's minor axis. Such a diameter allows clipping the fingers from the central region of the hand, as shown in Fig. 7 in the

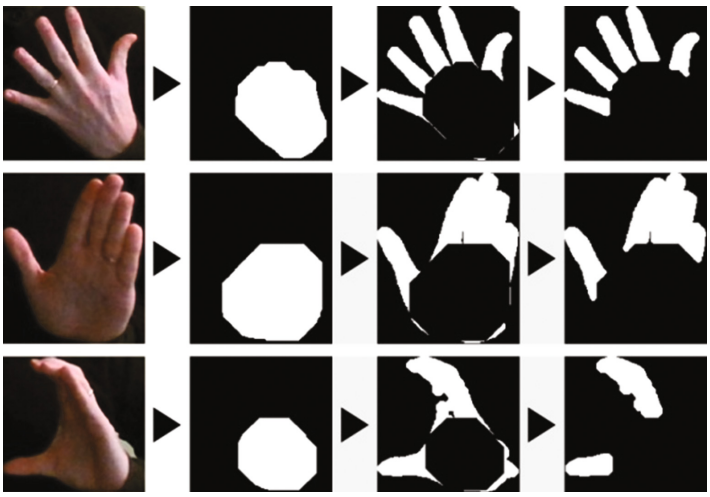

Fig. 7. Examples of clipping fingers from the central region of the hand in the image

second vertical row on the left. Then, a matrix, subjected to opening, should be excluded from the binary matrix with an object in the form of a hand. This procedure leads to separation of the central region of the hand from the fingers, as illustrated in Fig. 7 (the second column to the right).

As a result, we receive fingers in the form of objects with a preliminary removal of minor noises (as shown in Fig. 7 in the right column), for which the same features as for the hand are determined. This procedure allows us to have an idea about the hand as a whole and of its components in the form of fingers. The values of features are stored in the identifiers, the names of which do not coincide with each other. This allows using features in any order during recognition. Figure 8 shows a diagram of the method of calculating the visual features of hands.

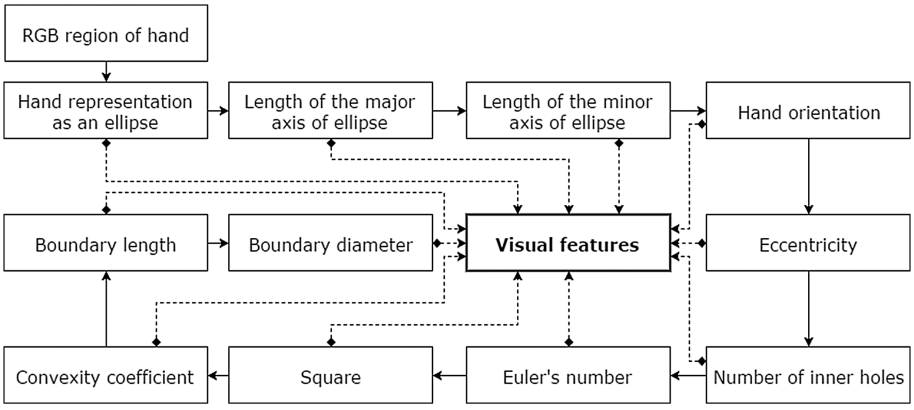

Fig. 8. Diagram of the method of calculating the visual features of one hand

Vectors of visual video features of hands and fingers are shown in Tables 1 and 2. The recognition process is as follows. If visual images are described using quantitative descriptors (length, square, diameter, texture, etc.), then the elements of decision theory can be applied.

Table 1. Visual features for describing a hand

Identifier	Feature description
hand_or	Hand orientation
hand_maj_axis_len	Length of the major axis of ellipse
hand_small_axis_len	Length of the minor axis of ellipse
hand_eccentr	Eccentricity
hand_open	Number of inner holes
hand_eul_num	Euler's number
hand_square	Square (area)
hand_convex	Convexity coefficient
hand_bord_len	Boundary length
hand_bord_diam	Boundary diameter

Table 2. Visual features for describing each finger

Identifier	Feature description
fing_or	Finger orientation
fing_maj_axis_len	Length of the major axis of ellipse
fing_small_axis_len	Length of the minor axis of ellipse
fing_eccentr	Eccentricity
fing_open	Number of inner holes
fing_eul_num	Euler's number
fing_square	Square (area)
fing_convex	Convexity coefficient
fing_bord_len	Boundary length
fing_bord_diam	Boundary diameter

Every gesture is represented as an image. An image is an arranged set of descriptors forming the feature vectors:

$$X = (x_1, x_2, \ldots, x_n)$$

where $x_i - i$ is a descriptor; n – total number of descriptors.

Images, which have some similar properties, form a class. In total, the recognition system comprises 52 classes (according to the number of recognized gestures in the database) referred to as $w_1, w_2, \ldots, w_{52}$. In each class, there are 5 records that are benchmarks for a proper showing of a certain gesture.

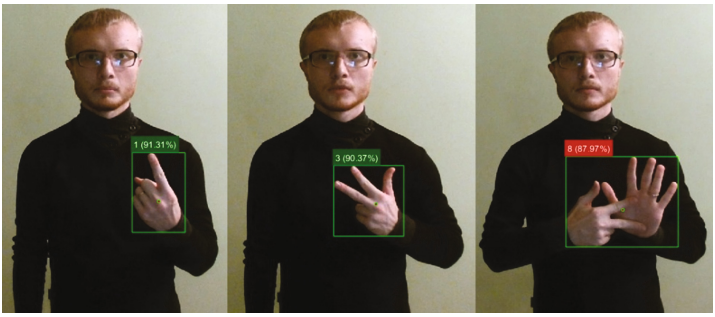

Fig. 9. Examples of gesture recognition

The process of matching the reference vectors with test ones is based on the calculation of the Euclidean distance between them. Belonging of an object to any class is based on the minimum distance between the reference (prototype) and unknown objects as shown in Fig. 9.

A static classifier by the minimum Euclidean distance is as follows. A reference class is the expectation of images vectors of the selected class:

$$m_j = \frac{1}{N}\sum_{x \in w_j} x_j$$

where $j = 1, 2, \ldots, W$ is the number of classes, N_j is the number of vectors of objects descriptors of the class w_j.

Summing is carried out over all vectors. A proximity measure based on the Euclidean distance is calculated according to the formula:

$$D_j(x) = \left\| x - m_j \right\|$$

where $j = 1, 2, \ldots, W$ is the number of classes, x is an unknown object, m_j is mathematical expectation calculated according to the previous formula.

Thus, the unknown object x will be correlated with a certain probability to such a class w_j, or which the proximity measure $D_j(x)$ will be the least.

Reaching the least (on average) error probability for the classification is carried out as follows. The probability that a certain object x belongs to the class w_j is $p(w_j|x)$. If the classifier refers an image (object) to the class w_j, which actually belongs to w_i, this indicates that there is a loss in the form of the classification error. These errors are referred to as L_{ij}. Since any image x в may be attributed to any of the existing classes W, the average loss value (error), associated with the attribution to the class w_j of the object x is equal to the average risk. The calculation is made according to the formula:

$$r_j(x) = \sum_{k=1}^{W} L_{kj} p(w_k|x)$$

Thus, an unknown image may be attributed to any class W. The sum of the average loss values over all admissible solutions will be minimal, because for the input image x the functions $r_1(x), r_2(x), \ldots, r_w(x)$ are calculated. This classifier is based on the Bayes classifier, i.e. it attributes the image x to the class w_i if $r_i(x) < r_j(x)$ on condition that $j = 1, 2, \ldots, W$ is the number of classes and the class j is not equal to the class i.

In our computer system, we employ some freely available software such as Open Source Computer Vision Library (OpenCV v3) [19], Open Graphics Library (OpenGL) [20] and Microsoft Kinect Software Development Kit (Kinect SDK 2.0) [21].

3 Experimental Research

The automated system was tested on different computers with various performance characteristics, parameters of which are presented in Table 3.

Table 3. The speed of frame processing with different computers

Processor	RAM, GB	Storage type	Video adapter	Processing speed, ms
Intel Core i7 3.4 GHz	8	HDD	Nvidia GeForce GTX 650 Ti	≈180
Intel Core i7 3.4 GHz	4	SSD	Nvidia GeForce GTX 650 Ti	≈140
Intel Atom Z250 1.33 GHz	2	HDD	GMA500	≈415
Intel Xeon E5-2690	128	SSD	NVIDIA Tesla K20X, NVIDIA Quadro K5000 (SLI)	≈70
Intel Core i5-3470	8	HDD	NVIDIA GeForce GT 640	≈270

The average processing time of one frame of the video sequence is 215 ms (Table 3), which allows processing up to 5 frames per 1 s. Current results do not allow processing video stream by an automation system in real time. However, there is a possibility of processing the recorded video fragments with the camera Kinect 2.0 together with the values obtained from the depth sensor in the synchronous mode.

The average recognition accuracy was 87%. Calculation is done according to the formula:

$$x_c = \frac{x_1 + x_2 + \ldots + x_n}{n},$$

where n is the number of gestures, x_n is the gesture recognition accuracy.

These results were obtained for the recorded database.

Gestures, for which it is necessary to put the fingers quite close to one another at different angles, showed the least recognition accuracy. In this case, the increasing of accuracy is possible by increasing the number of benchmarks and developing an

algorithm capable of performing the processes of opening and closing not only a binary image, but also color one. This will allow controlling color parameters flexibly and more accurately determine not only the coordinates of objects, but also their descriptors.

4 Conclusions and Future Work

In this paper, we presented the computer system aimed at recognition of manual gestures using Kinect 2.0. At present, our system is able to recognize continuous fingerspelling gestures and sequences of digits in Russian and Kazakh SLs. Our gesture vocabulary contains 52 isolated fingerspelling gestures. We have collected a visual database of SL gestures, which is available on request. This corpus is stored as a hierarchical database and consists of recordings of 2 persons. 5 samples of each gesture were applied for training models and the rest data were used for tuning and testing the developed recognition system. Feature vectors are extracted from both training and test samples of gestures, then comparison of reference patterns and sequences of test vectors is made using the Euclidian distance. Sequences of vectors are compared using the dynamic programming method and a reference pattern with a minimal distance is selected as a recognition result. According to our preliminary experiments in the signer-dependent mode with 2 demonstrators from the visual database, the average accuracy of gesture recognition is 87% for 52 manual signs.

In further research, we plan to apply statistical recognition techniques, e.g. based on some types of Hidden Markov Models (HMM) such as multi-stream or coupled HMMs, as well as deep learning approaches with deep neural networks [22]. We have plans to apply a high-speed video camera with >100 fps in order to keep dynamic gesture dynamics and to improve gesture recognition accuracy [23]. Also we plan to create an automatic lip-reading system for the full SL recognition system and to make a gesture-speech analysis [24] for SL. Our visual database should be extended with more gestures and signers to create a signer-independent system for RSL recognition. In future, this automatic SL recognition system will become a part of our universal assistive technology [25], including ambient assisted living environment [26, 27].

Acknowledgements. This research is partially supported by the Russian Foundation for Basic Research (project No. 16-37-60100), by the Council for Grants of the President of the Russian Federation (project No. MD-254.2017.8), by the state research (№ 0073-2014-0005), as well as by the Government of the Russian Federation (grant No. 074-U01).

References

1. Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Comput. Vis. Image Underst. **141**, 108–125 (2015)
2. Cooper, H., Ong, E.J., Pugeault, N., Bowden, R.: Sign language recognition using sub-units. J. Mach. Learn. Res. **13**, 2205–2231 (2012)

3. Guo, X., Yang, T.: Gesture recognition based on HMM-FNN model using a Kinect. J. Multimodal User Interfaces **11**, 1–7 (2016). doi:10.1007/s12193-016-0215-x. Springer
4. Karpov, A., Kipyatkova, I., Zelezny, M.: Automatic technologies for processing spoken sign languages. Procedia Comput. Sci. **81**, 201–207 (2016)
5. Karpov, A., Krnoul, Z., Zelezny, M., Ronzhin, A.: Multimodal synthesizer for Russian and Czech Sign Languages and Audio-Visual Speech. In: Stephanidis, C., Antona, M. (eds.) UAHCI/HCII 2013. LNCS, vol. 8009, pp. 520–529. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39188-0_56
6. Kindiroglu, A., Yalcin, H., Aran, O., Hruz, M., Campr, P., Akarun, L., Karpov, A.: Automatic recognition of fingerspelling gestures in multiple languages for a communication interface for the disabled. Pattern Recogn. Image Anal. **22**(4), 527–536 (2012)
7. Hruz, M., Campr, P., Dikici, E., Kindiroglu, A., Krnoul, Z., Ronzhin, A.L., Sak, H., Schorno, D., Akarun, L., Aran, O., Karpov, A., Saraclar, M., Zelezny, M.: Automatic fingersign to speech translation system. J. Multimodal User Interfaces **4**(2), 61–79 (2011)
8. Sousa, L., Rodrigues, J.M.F., Monteiro, J., Cardoso, P.J.S., Lam, R.: GyGSLA: a portable glove system for learning sign language alphabet. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2016. LNCS, vol. 9739, pp. 159–170. Springer, Cham (2016). doi:10.1007/978-3-319-40238-3_16
9. Shibata, H., Nishimura, H., Tanaka, H.: Basic investigation for improvement of sign language recognition using classification scheme. In: Yamamoto, S. (ed.) HIMI 2016. LNCS, vol. 9734, pp. 563–574. Springer, Cham (2016). doi:10.1007/978-3-319-40349-6_55
10. Nagashima, Y., et al.: A support tool for analyzing the 3D motions of sign language and the construction of a morpheme dictionary. In: Stephanidis, C. (ed.) HCI 2016. CCIS, vol. 618, pp. 124–129. Springer, Cham (2016). doi:10.1007/978-3-319-40542-1_20
11. Sako, S., Hatano, M., Kitamura, T.: Real-time Japanese sign language recognition based on three phonological elements of sign. In: Stephanidis, C. (ed.) HCI 2016. CCIS, vol. 618, pp. 130–136. Springer, Cham (2016). doi:10.1007/978-3-319-40542-1_21
12. Halim, Z., Abbas, G.: A Kinect-based sign language hand gesture recognition system for hearing- and speech-impaired: a pilot study of Pakistani sign language. Assistive Technol. **27** (1), 34–43 (2015)
13. Chong, W., Zhong, L., Shing-Chow, C.: Superpixel-based hand gesture recognition with Kinect depth camera. IEEE Trans. Multimed. **1**(17), 29–39 (2015)
14. Microsoft Developer Network. Skeletal Tracking. https://msdn.microsoft.com/en-us/library/hh973074.aspx
15. Sharma, D., Vatta, S.: Optimizing the search in hierarchical database using Quad Tree. Int. J. Sci. Res. Sci. Eng. Technol. **1**(4), 221–226 (2015). Springer
16. Sreedhar, K., Panlal, B.: Enhancement of images using morphological transformations. Int. J. Comput. Sci. Inf. Technol. **4**(1), 33–50 (2012)
17. Sossa-Azuela, J.H., Santiago-Montero, R., Pérez-Cisneros, M., Rubio-Espino, E.: Computing the Euler number of a binary image based on a vertex codification. J. Appl. Res. Technology. **11**, 360–370 (2013)
18. Chaple G., Daruwala R., Gofane, M.: Comparisons of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA. In: Proceeding International Conference on Technologies for Sustainable Development ICTSD-2015. IEEEXplore (2015)
19. Kaehler, A., Bradsky, G.: Learning OpenCV 3. O'Reilly Media, California (2017)
20. OpenGL library. https://www.opengl.org
21. Kinect for Windows SDK 2.0. https://www.microsoft.com/en-us/download/details.aspx?id=44561
22. Kipyatkova, I.S., Karpov, A.A.: Variants of deep artificial neural networks for speech recognition systems. SPIIRAS Proc. **49**(6), 80–103 (2016). doi:10.15622/sp.49.5

23. Ivanko, D.V., Karpov, A.A.: An analysis of perspectives for using high-speed cameras in processing dynamic video information. SPIIRAS Proc. **44**(1), 98–113 (2016). doi:10.15622/sp.44.7
24. Sargin, M., Aran, O., Karpov, A., Ofli, F., Yasinnik, Y., Wilson, S., Erzin, E., Yemez, Y., Tekalp, M.: Combined gesture-speech analysis and speech driven gesture synthesis. In: Proceeding IEEE International Conference on Multimedia and Expo ICME-2006, Toronto, Canada. IEEEXplore (2006)
25. Karpov, A., Ronzhin, A.: A universal assistive technology with multimodal input and multimedia output interfaces. In: Stephanidis, C., Antona, M. (eds.) UAHCI/HCII 2014. LNCS, vol. 8513, pp. 369–378. Springer, Cham (2014). doi:10.1007/978-3-319-07437-5_35
26. Karpov, A., Akarun, L., Yalçın, H., Ronzhin, A.L., Demiröz B., Çoban A., Zelezny M.: Audio-visual signal processing in a multimodal assisted living environment. In: Proceeding of 15th International Conference INTERSPEECH-2014, Singapore, pp. 1023–1027 (2014)
27. Karpov, A., Ronzhin, A., Kipyatkova, I.: Automatic analysis of speech and acoustic events for ambient assisted living. In: Antona, M., Stephanidis, C. (eds.) UAHCI/HCII 2015. LNCS, vol. 9176, pp. 455–463. Springer, Cham (2015). doi:10.1007/978-3-319-20681-3_43