Sign Language Recognition Based on Trajectory Modeling with HMMs

Junfu Pu, Wengang Zhou(✉), Jihai Zhang, and Houqiang Li

University of Science and Technology of China, Hefei, Anhui,
People's Republic of China
{pjh,jihzhang}@mail.ustc.edu.cn, {zhwg,lihq}@ustc.edu.cn

Abstract. Sign language recognition targets on interpreting and understanding the sign language for convenience of communication between the deaf and the normal people, which has broad social impact. The problem is challenging due to the large variations for different signers and the subtle difference between sign words. In this paper, we propose a new method for isolated sign language recognition based on trajectory modeling with hidden Markov models (HMMs). In our approach, we first normalize and re-sample the raw trajectory data and partition the trajectory into multiple segments. To represent each trajectory segment, we proposed a new curve feature descriptor based on shape context. After that, hidden Markov model is used to model each isolated sign word for recognition. To evaluate the performance of our proposed algorithm, we have built a large isolated Chinese sign language vocabulary with Kinect 2.0. The dataset contains 100 unique isolated sign words, each of which is performed by 50 signers for 5 times. Experimental results demonstrate that the proposed method achieves a better performance compared with normal coordinate feature with HMM.

1 Introduction

Sign language is one of the most important ways for communication between the deaf and the normal people. With broad social impact, this problem has attracted considerable attention from many researchers around the world. The target is to build a system to automatically translate sign language into text or interpret it into spoken language [17,18]. The research methods of sign language recognition (SLR) can also be applied in general human-computer interaction systems.

Sign language conveys semantic meaning through hand shapes, trajectories, and facial expressions. Most existing recognition methods for sign language are based on gestures and trajectories of sign words. For gesture recognition, Murakami and Taguchi [9] proposed a gesture recognition method for Japanese sign language using recurrent neural network, but they used the data gloves, which were expensive and inconvenient in real application for the signers. They developed a posture recognition system which could recognize a finger alphabet of 42 symbols and achieved a recognition rate of 98 % for registered people.

Q. Tian et al. (Eds.): MMM 2016, Part I, LNCS 9516, pp. 686–697, 2016.
DOI: 10.1007/978-3-319-27671-7_58

Also, an alphabet gesture recognition system was designed for American sign language (ASL) with ANN by Oz and Leu [10]. Hidden Markov models (HMMs) [13] show an extremely good performance in temporal pattern recognition, especially in speech recognition [2,12]. Schlenzig et al. [14] used a single universal HMM and a finite state estimator for the determination of gestures. Their proposed method also achieved a high recognition rate. Huang et al. [6] built a deep neural network based on postures captured by Real-Sense.

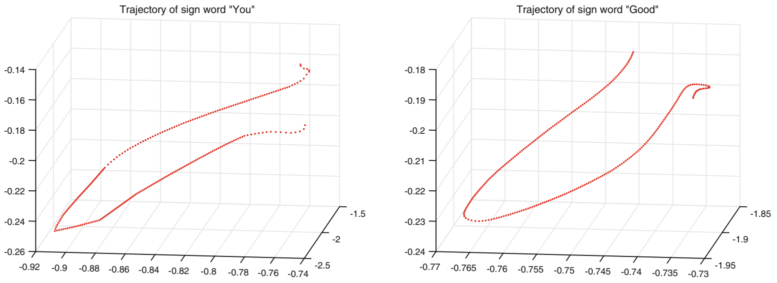
The above works are commonly based on gesture recognition. However, these methods are not able to handle the recognition for signs with only trajectory information. To address this problem, more and more researchers focus on sign language recognition by trajectory matching [16]. Yushun et al. [8] presented a new method of curve matching from the views of manifold for sign language recognition. They divided the curve of the sign word into a set of several linear segments, and defined the distance between two segments. Thus, the matching of two curves was transformed into the matching between two sets of linear segments. Their method achieved a recognition rate of 78.3 % in a dataset with 370 daily sign words. In addition, the combination of both trajectory and hand shape features for sign language recognition was proposed by Grobel and Assan [3], and data gloves were also necessary in there experiments. Although sign language recognition with data gloves [4] achieved a high recognition rate, it's inconvenient to be applied in SLR system for the expensive device. Kinect developed by Microsoft [15] is capable of capturing the depth, color, and joint locations easily and accurately. Hence, more and more researchers use Kinect for sign language recognition [5,20–22].

The method we proposed in this paper is based on trajectories of sign words. The data captured by Kinect consists of a set of 3D points, which are the axis locations of joints in each temporal stamp. We use the trajectories of both hands for recognition. However, recognition only based on trajectories suffers difficulties in some specific cases. Take the Chinese sign words "You" and "Good" for example, the trajectories of both words are shown in Fig. 1. These two trajectories are quite similar, which makes it tough to realize the similar signs' accurate recognition.

The rest of this paper is organized as follows. Section 2 gives an overview of our system. The method of feature extraction is discussed in Sect. 3. Section 4 introduces the details of sign words modeling with HMMs. The experiments are carried out in Sect. 5. Finally, in Sect. 6 we make a summary and brief discussion for future work.

2 System Overview

As shown in Fig. 2, the trajectory matching system for sign language recognition consists of 5 modules, i.e., data preprocessing, discrete contour evolution (DCE) [7] for segmentation, feature extraction for HMMs, codebook training and quantization, and trajectory matching with HMMs. The aim of preprocessing is to normalize the data and decrease the negative effect of noise. We re-sample the



(a) Sign word "You"

(b) Sign word "Good"

Fig. 1. The illustration of two trajectories for Chinese sign words "You" and "Good". These two trajectory curves are quite similar with subtle difference.

raw data path into a new curve with fixed length. For the re-sampled curves, the DCE algorithm [7] is used for segmentation. Then we extract the shape context [1] of each point and randomly choose some of them for training a codebook. The curve features introduced in Sect. 3 are extracted for each trajectory. With these features, we build a HMM model for each isolated sign word. Then the label with maximum posterior probability is regarded as the recognition result.

3 Curve Feature Extraction

In this section, we will introduce how to build robust features with an effective representation of the 3D trajectory curve. These features will be used in trajectory curve recognition and retrieval. First of all, we extract the shape context of all points in the curve. For all word samples, we choose some of them randomly for training codebooks, without taking account of their labels. Then for each word sample, we quantize the shape context features of all points in the trajectory curve with the pre-trained codebooks. An illustration of curve feature extraction based on shape context is shown in Fig. 3.

3.1 Shape Context

In literature, shape context [1] has been wildly used in shape recognition. Shape context is a feature that describes the distribution of other points in the neighborhood of a reference point. For a point p_i on the curve, shape context is defined as a histogram h_i of the relative coordinates of the remaining $n-1$ points, where n is the number of points in the curve. Equation 1 gives the definition of h_i, where $\#$ means the cardinality of a set. In order to make the descriptor more discriminative to nearby points, we use $log-polar^2$ coordinate system. A brief illustration is shown in Fig. 4.

$$h_i(k) = \#\{q \neq p_i : \ (q - p_i) \in bin(k)\}. \tag{1}$$

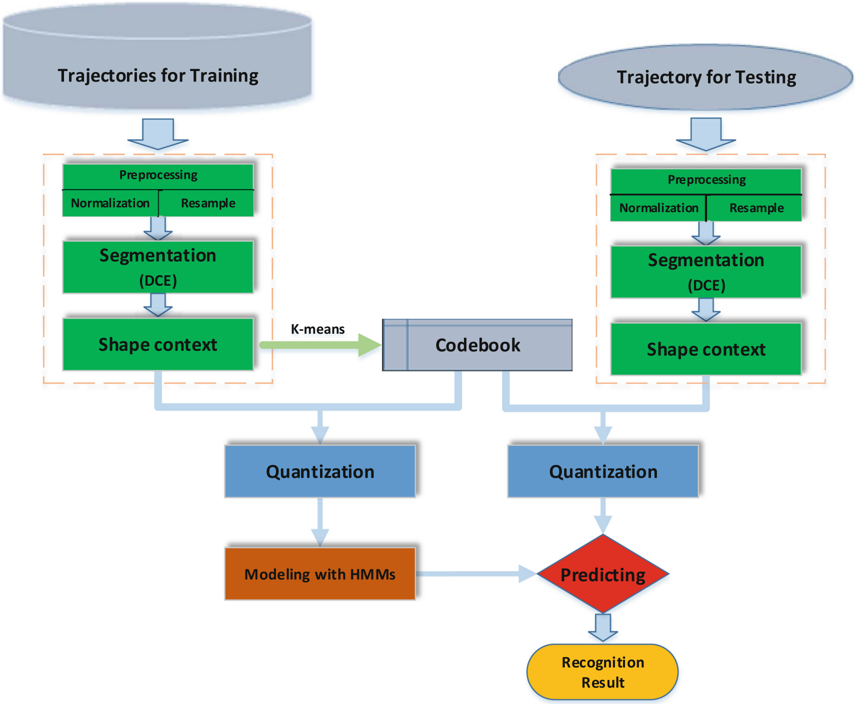


Fig. 2. An illustration of proposed method. The steps in the left column consist the stage of training. For a testing trajectory of sign word, the HMMs that we've trained are used for recognition.

Considering that our sign word trajectory is 3D, we project it along three orthogonal planes, i.e., x-y, y-z, and x-z, and obtain three 2D curves. Then, for each curve, we extract a shape context feature and concatenate them into a single feature vector to represent the 3D sign word trajectory. We use 2D shape context feature since they are efficient for extraction and well capture the data structure.

3.2 Codebook Training

After getting the shape context features of all sample points in a sign word trajectory, we train the codebook for quantization. Here, we use K-means algorithm for codebook training. The main steps are as follow: First, we randomly sample a set of training features. Then we extract shape context features of these chosen samples and use K-means clustering algorithm to generate a set of cluster centers. The cluster centroids constitute our codebook. Suppose the codebook size is K, then the codebook can be described as $B = [b_1, b_2, \ldots, b_K]^T \in R^{K\times d}$, where b_i is the cluster centroid vector and d denotes the dimension of shape context feature.

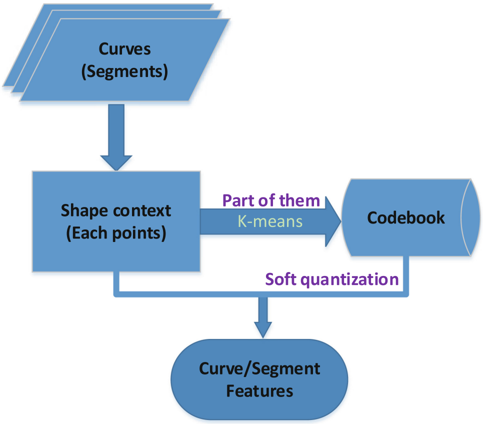


Fig. 3. The flow chart of curve feature extraction

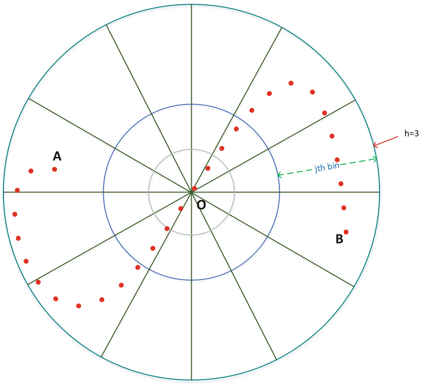


Fig. 4. An illustration of shape context. For point O, we count the points dropped into each bin. As is marked in this figure, there are 3 points in the j-th bin, so $h_j = 3$.

3.3 Quantization

A 3D trajectory is formed with a set of sequential 3D points. For a N-point curve $C = (p_1, p_2, \cdots, p_N)$, we use the codebook discussed in Sect. 3.2 to quantize the shape context feature of each point p_i. In our experiment, soft quantization [11] is used to get curve feature. The illustration of soft quantization is shown in Fig. 5.

As is shown in Fig. 5, $B = [b_1, b_2, \ldots, b_K]^T$ is the codebook, which b_i is the cluster centroid feature. SC_i denotes the shape context of point p_i in curve C. We calculate the Euclidean distance between SC_i and $b_q (q = 1, 2, \cdots, K)$.

$$d_{i,q} = ||SC_i - b_q||. \tag{2}$$

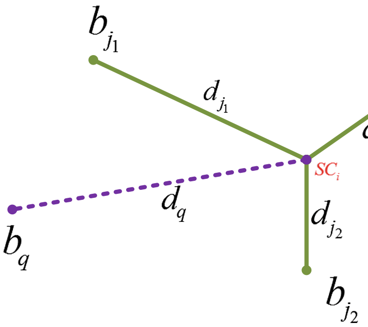


Fig. 5. An illustration for soft-quantization. b_{j_1}, b_{j_2} and b_{j_3} are 3 nearest centroid features far from SC_i in feature space. So, we use the distance d_{j_1}, d_{j_2} and d_{j_3} to calculate the weights for quantization.

Denote $d_i = [d_{i,1}, d_{i,2}, \ldots, d_{i,k}]^T$, then we select Q smallest distances $d_{j_1}, d_{j_2}, \ldots, d_{j_Q}$, and the weights of quantization is given by Eq. 3 [11].

$$\omega_i = \frac{exp^{-\frac{d_i^2}{\sigma^2}}}{\sum\limits_{j=1}^{Q} exp^{-\frac{d_j^2}{\sigma^2}}}, \quad i = 1, 2, \cdots, Q, \tag{3}$$

where σ is a constant. Then the curve feature $f = [f_1, f_2, \ldots, f_K]$ could be generated as Algorithm 1.

Algorithm 1. Extraction of Curve Feature

```
Input: The 3D trajectory points $C = (p_1, p_2, \cdots, p_N)$;
    Codebook $B = [b_1, b_2, \cdots, b_K]^T \in R^{K \times d}$;
Output: The curve feature with a good enough description of curve $C$;
n=1;
$f = [f_1, f_2, \cdots, f_K]^{K \times 1}$, set $f_i = 0$ for all $i = 1, 2, \cdots, K$;
for $n = 1 : Q$ do
    Extract the shape context $SC_n$ of point $p_n$;
    $d = [d_1, d_2, \cdots, d_K]^{K \times 1}, d = B \times SC_n$;
    Make $d$ sorted by increasing and get Q minimum values $d_{i_1}, d_{i_2}, \cdots, d_{i_Q}$ and its
    corresponding index $idx_{i_1}, idx_{i_2}, \cdots, idx_{i_Q}$;
    for $j = 1 \rightarrow Q$ do
        Get $\omega_j$ refers to Eq. 3;
        $f_{idx_j} = f_{idx_j} + \omega_j$;
    end for
end for
Output the curve feature $f$;
```

4 Character Modeling by HMM

Hidden Markov models are well known for their application in temporal patten recognition such as speech and handwritten characters. In sign language recognition, the trajectories can also be regarded as temporal sequences, and HMMs are appropriate for dealing with this problem.

Let's denote $C = (p_1, p_2, \ldots, p_N)$ as the whole curve with N points, which is divided into M segments. Suppose $C^{(i)}(i = 1, 2, \cdots, M)$ denotes the segment set of C, then $C = \bigcup_{i=1}^{M} C^{(i)}(i = 1, 2, \ldots, M)$. Actually, $C^{(i)}$ can be regarded as a set of sub-motions and the context between these sub-motions is going to be modeled with HMMs. In our experiments, we use DCE (Discrete Contour Evolution) algorithm to partition the curve path into M segments. Data preprocessing is necessary before segmentation.

4.1 Preprocessing

The procedure of data preprocessing includes two aspects: normalization and resampling. The trajectories should be normalized since they are quite different in scales when performed by different signers. Specifically, we use the location of the signer's head and width of shoulder to realize normalization. Resampling also plays an important role in preprocessing. It will make the trajectories more smooth and remove the noise. Generally, the velocities of different parts can be much different when a signer is playing a sign language word, so the sample points are not uniformly distributed, resampling will solve this problem much more better. We use the $1 algorithm proposed by Wobbrock et al. [19] for resampling.

4.2 DCE Algorithm

In sign language recognition, $C = (p_1, p_2, \ldots, p_N)$ is composed of digital line segments $s_1, s_2, \ldots, s_{N-1}$, where s_i is the line segment joining p_i to p_{i+1} for $i = 1, 2, \ldots, N-1$. We normalize s_i by the length of curve C. Let's denote $l(s)$ as the length of s, and $\beta(s_1, s_2)$ as the angle between s_1 and s_2. The cost function $Z(s_1, s_2)$ for a pair of segments s_1 and s_2 is defined in Eq. 4 [7].

$$Z(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)}. \tag{4}$$

The main procedure of DCE is given by Algorithm 2 [7].

Algorithm 2. DCE Algorithm

```
Input: The 3D trajectory points $C = (p_1, p_2, ..., p_N)$;
    The number of segments $M$;
Output: The segmentation $C^{(i)}(i = 1, 2, ..., M)$ of curve $C$;
k=N;
while k>M+1 do
    Find a pair $s_i, s_{i+1}$ such that $Z(s_i, s_{i+1})$ is minimal;
    Replace $s_i, s_{i+1}$ by the segment $s^{'}$ joining the endpoints of arc $s_i \bigcup s_{i+1}$;
    k=k-1;
end while
Use the index of remaining points to get segmentation $C^{(i)}, i = 1, 2, ..., M$;
```

4.3 HMM Modeling

After getting the partition of curve C, we can extract curve feature from each sub-motion $C^{(i)}(i = 1, 2, \ldots, M)$ for HMMs training. For each word, we can use some of these curve features as training samples to build a hidden Markov model. Each character HMM is structured as left-to-right model.

The HMM is modeled by the parameter vector $\lambda = (A, B, \pi)$, where A is the transition probability matrix, B is the emission probability matrix, and π is the initial state probability distribution. For each word, we can use our training samples to get the parameter vector λ by Baum-Welch algorithm. Suppose there are N HMMs to be trained, that is to say, the whole data set contains N different isolated sign language words. Given an unknown testing sequence $O = [o_1, o_2, \ldots, o_n]$, we classify it to class C_p with the following decision rule:

$$C_p = \max_{C_i} \log p(O|\lambda_{C_i}), \quad i = 1, 2, \ldots, N, \tag{5}$$

where $\log p(O|\lambda)$ is the logarithm of the probability of sequence O, given the model parameter λ. The sum of $\log p(O|\lambda)$ for both hands is used for recognition.

5 Experiments

5.1 Datasets and Experimental Setup

Our dataset is built by ourselves with Kinect and will be released to the public. It contains 100 isolated Chinese sign language words in daily life. Each word is played by 50 signers for 5 times. As a result, the dataset consists of $100 \times 50 \times 5$ samples. We divide the whole dataset into 2 subsets for training, validation, and testing. The details about these 2 subsets are shown as follow:

Subset A. Subset A contains 100 words, each of which is performed by 14 signers for 5 times. So there are 7,000 samples in total. In the experiments introduced in Sect. 5.2, we choose 60 samples of each word for training, and the rest 10 for validation.

Subset B. The vocabularies for both dataset are the same. Each word in Subset B is performed by another 36 signers for 5 times. In our experiments, we use Subset B as a large testing set to evaluate the effectiveness and stability of our method.

The Kinect developed by Microsoft is able to detect the joints and we can obtain the locations of joints in real-time. In sign language recognition, the locations of both hands will make sense for recognition. Hence, the trajectories of left hand and right hand will be modeled by HMM independently.

5.2 Optimal Parameters Setting

For the step of preprocessing, we re-sample the raw trajectory into a new path with 300 points. Suppose there are V observations and Q hidden states, and the trajectory is divided into M segments. It's tough to determine the optimal parameters at the same time, making the models fit the sign words well. So, we vary one parameter with the other parameters fixed. Hence, we can get approximate optimal parameters by experiments in this way. For each word, we choose 60 samples of each word in Subset A for training and the rest in Subset A for validation.

As shown in Fig. 6, when we fixed the segment number M as 15, the optimal V and Q are 10 and 8, respectively, and it can reach the accuracy rate up to 60.1 %. Hence, it reasonable to use the optimal parameters in the final recognition. In the same way, we can fix the V and Q randomly and find the optimal segment number. The results with fixed $V = 20$ and $Q = 6$ are shown in Fig. 7. We can find that when we choose the segment number as 30, we obtain the best performance. Hence, in the following test, we set the parameters as $V = 10$, $Q = 8$ and $M = 30$.

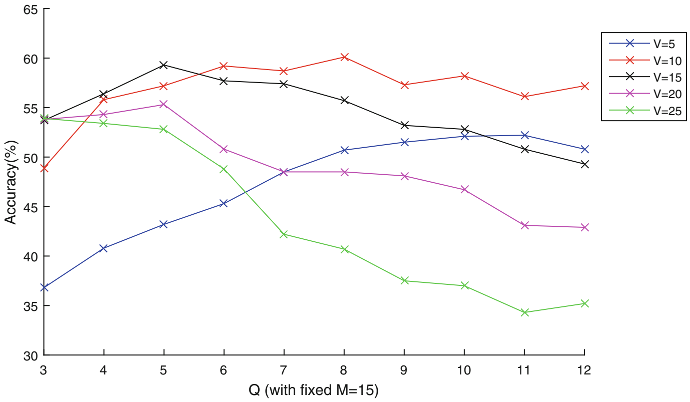


Fig. 6. Recognition rates with various V and Q. We fixed the number of segments as 15.

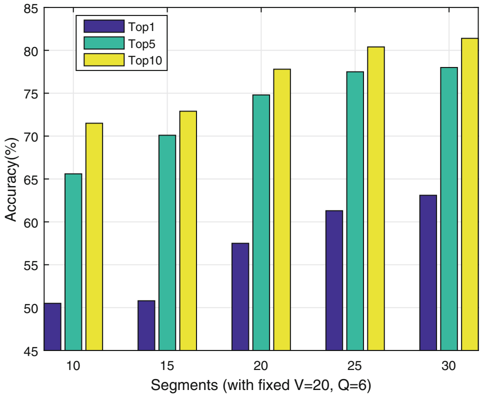


Fig. 7. Recognition rates with virous M. With fixed $V = 20$ and $Q = 6$, we find that $M = 30$ is the optimal parameter.

5.3 Results and Analysis

After getting the optimal parameters of V, Q and M, we use the dataset built by ourselves to evaluate the performance of our method. HMM with normal coordinate and curve matching from the view of manifold (CM_VoM) proposed in [8] are our baseline methods. In our experiments with Subset A, we choose 60 samples of each word for training while the rest 10 for testing. The recognition rates on Subset A for different methods are shown in Table 1. Our method can get the accuracies of 67.3 %, 86.6 %, 89.8 % in top 1, top 5, and top 10, respectively. While HMM with normal coordinate features gets the accuracies of 32.2 %, 54.8 %, 65.3 % in top 1, top 5 and top 10, respectively. CM_VoM [8] obtains the accuracies of 57.6 %, 82.4 %, 89.4 %, respectively. In order to show the effectiveness and stability of our method, we also conduct the experiments on large Subset B. We use the pre-trained models with Subset A to recognize the words in Subset B. That is to say, the training and testing samples are from different signers. Table 2 gives the accuracies of different methods on Subset B. Our method can get the recognition rates at 54.4 %, 77.3 %, and 82.7 % in top 1, top 5, top 10, respectively, which performs better than the other 2 methods.

Table 1. The recognition rates for different methods on Subset A

Subset A	Top1	Top5	Top10
Normal HMM	0.322	0.548	0.653
CM_VoM	0.576	0.824	0.894
Our method	0.673	0.866	0.898

Table 2. The recognition rates for different methods on Subset B with unseen signers

Subset B	Top1	Top5	Top10
Normal HMM	0.125	0.271	0.386
CM_VoM	0.451	0.719	0.819
Our method	0.544	0.773	0.827

6 Conclusion

In this paper, we propose a new approach for Chinese sign language recognition based on trajectory modeling. The method is inspired from the shape recognition with shape context. We partition the projected curve of sign word trajectory into multiple segments and represent each segment into histogram feature by shape context quantization. With these features, the HMMs are applied for modeling the sign words. The experiments show that our method outperforms the comparison methods by a large margin on a large dataset containing 100 sign words with over 25,000 samples. For the future work, we will integrate both trajectory of sign word and hand shapes for more accurate SLR recognition.

Acknowledgement. This work was supported in part to Dr. Zhou by the Fundamental Research Funds for the Central Universities under contract No. WK2100060014 and WK2100060011 and the National Science Foundation of China under contract No. 61472378, and in part to Prof. Li by the National Science Foundation of China under contract No. 61272316.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)
2. Gales, M., Young, S.: The application of hidden markov models in speech recognition. Found. Trends Sig. Process. **1**(3), 195–304 (2008)
3. Grobel, K., Assan, M.: Isolated sign language recognition using hidden markov models. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 162–167. IEEE (1997)
4. Hienz, H., Kraiss, K.-F., Bauer, B.: Continuous sign language recognition using hidden markov models. In: International Conference on Multimodal Interfaces, vol. 4, pp. 10–15 (1999)
5. Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using 3D convolutional neural networks. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE, 2015
6. Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using real-sense. In: IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp. 166–170. IEEE (2015)
7. Latecki, L.J., Lakämper, R.: Convexity rule for shape decomposition based on discrete contour evolution. Comput. Vis. Image Underst. **73**(3), 441–454 (1999)

8. Lin, Y., Chai, X., Zhou, Y., Chen, X.: Curve matching from the view of manifold for sign language recognition. In: Shan, S., Jawahar, C.V., Jawahar, C.V. (eds.) ACCV 2014 Workshops. LNCS, vol. 9010, pp. 233–246. Springer, Heidelberg (2014)
9. Murakami, K., Taguchi, H.: Gesture recognition using recurrent neural networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 237–242. ACM (1991)
10. Oz, C., Leu, M.C.: Recognition of finger spelling of American sign language with artificial neural network using position/orientation sensors and data glove. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3497, pp. 157–164. Springer, Heidelberg (2005)
11. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
12. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
13. Rabiner, L.R., Juang, B.-H.: An introduction to hidden markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
14. Schlenzig, J., Hunter, E., Jain, R.: Vision based hand gesture interpretation using recursive estimation. In: Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1267–1271. IEEE (1994)
15. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Commun. ACM **56**(1), 116–124 (2013)
16. Wang, H., Chai, X., Zhou, Y., Chen, X.: Fast sign language recognition benefited from low rank approximation. In: IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, pp. 1–6. IEEE (2015)
17. Wang, M., Hua, X.-S., Hong, R., Tang, J., Qi, G.-J., Song, Y.: Unified video annotation via multigraph learning. IEEE Trans. Circuits Syst. Video Technol. **19**(5), 733–746 (2009)
18. Wang, M., Ni, B., Hua, X.-S., Chua, T.-S.: Assistive tagging: a survey of multimedia tagging with human-computer joint exploration. ACM Comput. Surv. (CSUR) **44**(4), 25 (2012)
19. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, pp. 159–168. ACM (2007)
20. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P.: American sign language recognition with the kinect. In: Proceedings of the 13th International Conference on Multimodal Interfaces, pp. 279–286. ACM (2011)
21. Zhang, J., Zhou, W., Li, H.: A threshold-based hmm-dtw approach for continuous sign language recognition. In: Proceedings of International Conference on Internet Multimedia Computing and Service, p. 237. ACM (2014)
22. Zhang, J., Zhou, W., Li, H.: A new system for chinese sign language recognition. In: IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp. 534–538. IEEE (2015)