J Real-Time Image Proc
DOI 10.1007/s11554-016-0620-0

SPECIAL ISSUE PAPER

Spatial and temporal pyramid-based real-time gesture recognition

Feng Jiang[1] · Jie Ren[1] · Changhoon Lee[2] · Wuzhen Shi[1] · Shaohui Liu[1] · Debin Zhao[1]

Received: 29 December 2015 / Accepted: 7 July 2016

Abstract This paper proposes a novel method for real-time gesture recognition. Aiming at improving the effectiveness and accuracy of HGR, spatial pyramid is applied to linguistically segment gesture sequence into linguistic units and a temporal pyramid is proposed to get a time-related histogram for each single gesture. Those two pyramids can help to extract more comprehensive information of human gestures from RGB and depth video. A two-layered HGR is further exploited to further reduce the computation complexity. The proposed method obtains high accuracy and low computation complexity performance on the ChaLearn Gesture Dataset, comprising more than 50, 000 gesture sequences recorded.

Keywords Spatial pyramid · Temporal pyramid · One-shot learning · Gesture recognition · DTW

1 Introduction

Gestures are the unsaid words of human beings, which are expressed in the form of actions [1]. They are considered as the most natural expressive means of communication between human beings and computers in a virtual system [2]. For the purpose of improving human beings' interaction with machines, considerable scholarly work has been undertaken on HGR whose extensive applications include sign language recognition [3], socially assistive robotics [4], directional indication through pointing [5], immersive gaming technology, virtual controllers, and remote control.

Hand gesture recognition (HGR) is of great importance owing to the hand's freer expressions than other body parts [6]. Although hand postures and gestures are frequently considered as being identical, there are marked differences as explained in [7]. The hand posture is a static motionless pose, such as making a palm posture and holding it in a certain position, while the hand gesture is a dynamic process consisting of a sequence of changing hand postures over a short duration. Hand gestures are powerful human–human communication interface components. However, their fluency and intuitiveness have not been extensively utilized for human–computer communication interface. The techniques are still too fragile or too coarse grained to be of any universal use for HGR. They still lack robustness and ability to recognize gestures in a convenient and easily accessible manner. Therefore, the more efficient techniques for human–computer interface with hand gestures should be developed.

In the literature, many researchers have tried to study HGR with different equipments [8, 9]. These sensor-based methods have high recognition accuracy because they can precisely catch the movement of hands. However, due to the high cost of the sensors used, these techniques are impractical in real applications [10]. Consequently, computer vision-based HGR, which can perform recognition as naturally as in human–human interaction, has been considered to be the most promising method. Generally, numerous phases and imaging restrictions and other constraints are needed to get better performance [8]. The well-known "come as you are" expression [11] stands for a human–computer communication interface without constraints, such as the user using markers, color gloves, or

✉ Feng Jiang
fjiang@hit.edu.cn

[1] School of Computer Science, Harbin Institute of Technology, Harbin, China

[2] Seoul National University of Science and Technology, Seoul, Korea

Published online: 18 July 2016

sleeves, and the environment having uniform background. It is a challenge problem for computer vision-based HGR to get better performance without these constraints.

1.1 Hand gesture recognition with Kinect devices

3D sensors have been thought as the key to resolving gesture recognition problems. Depth sensors, multiple camera systems, and 3D scanners have been used to recognize gesture in 3D space [12–15]. However, due to high equipment cost, inconvenience to use, and the complexity of modeling, they have not become the mainstream devices for gesture recognition.

In June 2011, Microsoft released a Kinect Software Development Kit (SDK), which includes a set of powerful algorithms for extracting scene depth and object silhouettes and subsequently building a skeleton model of a person in front of the camera in real time. Kinect's target application was video games; however, its 3-D information capture capabilities have helped spawn numerous research projects in the field of human–computer interfaces. Although its resolution and accuracy are low, its low cost promises to make it the primary 3D information capturing device in gesture recognition. Some work has been done related to Kinect-based gesture recognition and other similar areas [16, 17]. Usually, a set of parameters, such as thresholds on joint locations, velocities, and accelerations, is specified to localize and track movements. Accordingly, a number of applications have been developed using Flexible Action and Articulated Skeleton Toolkit (FAAST) [18].

The challenge of HGR lies in the understanding of the unique characteristics and cues of the gestures themselves. Drawing inspiration from these cues and organizing the recognition algorithms accordingly are crucial steps toward the improved HGR performance. The proposed spatial and temporal pyramid-based real-time gesture recognition takes into account both the simultaneous and sequential organization of gestures. Aiming at improving the effectiveness and accuracy of HGR, spatial pyramid is applied to linguistically segment gesture sequence into linguistic units and a temporal pyramid is proposed to get a time-related histogram for each single gesture. Those two pyramids can help to extract more comprehensive information of human gestures from RGB and depth video. A two-layered HGR is further exploited to further reduce the computation complexity. The overall framework that integrates all of the above is evaluated on data from ChaLearn Gesture Dataset (CGD2011) [19].

The remainder of the paper is organized as follows: Related works are presented in Section II. Section III elaborates the architecture and detailed implementation of two-layered HGR. Extensive experimental results are reported in Section IV and finally in Section V, we summarize this paper.

2 Related work

In vision-based HGR, a set of features is extracted from every frame, and then, a classifier is applied to recognize different gestures. The vision-based HGR can be divided into two categories, namely the three-dimensional (3D) hand model-based methods and the appearance-based methods [20]. 3D model-based methods provide a geometrical representation of the hand configuration using the joint angles of a 3D hand's articulated structure recovered from the sequence of images [21]. The 3D hand model-based technique provides a rich description that permits a wide range of hand gestures. However, since the 3D hand models are articulated deformable objects with high degree of freedom, a large image database is required to deal with the entire characteristic shapes under several views. Appearance-based techniques extract image features to model the visual appearance of the gesture and then compare these features with pattern classification module [22], which is widely used in vision-based HGR.

2.1 Feature extraction

Feature extraction is intended for collecting data about gesture position, orientation, and temporal progression. In early works, the feature extraction is simplified considerably by requiring the subjects to wear single or differently colored gloves [23, 24]. Using colored gloves reduces the encumbrance to the gesturer but does not remove it completely.

A more natural, realistic approach is not to use color gloves. The most common approach extracts the features with skin color model [25–28] where a common restriction is the wearing of long-sleeved clothes. Often feature extraction is simplified by restricting the background to a specific color or at the very least keeping it uncluttered and static [16]. Akyol and Alvarado [29] improved the original color-based skin segmented tracker by using a combination of skin segmentation and MHIs to detect the hands. Awad et al. [25] presented a face and hand tracking system that combined skin segmentation, frame differencing, and predicted positioning in a probabilistic manner. These reduced the confusion with static background images but continued to suffer problems associated with bare forearms.

There are other methods used in the gesture feature extraction. Wong et al. [30] used PCA on motion gradient images of a sequence to obtain features for a Bayesian classifier. In [31], Zahedi et al. used intensity images, skin color images, and different first- and second-order

derivative features to recognize words of American Sign Language, and good results are achieved. Cooper et al. [10] proposed a method for sign gesture recognition on a small sign subset that bypasses the need for tracking entirely. They classified the motion directly by using volumetric Haar-like features in the spatiotemporal domain and demonstrated that non-tracking-based approaches can also be used at the sub-unit level.

Depth information from a calibrated camera pair or direct depth sensors such as Light Detection and Ranging (LiDAR) is a good indication. If the person is expected to face the camera(s), the closest objects may be considered as the hands. Many tentative works have been done with Kinect [16, 17]. The skeleton information offered by Kinect is more effective in the expression of human action than pure depth data. However, there are a lot of cases that skeleton cannot work such as interaction between human body and other object (e.g., moving a box) and micro-movement in close distance (e.g., making a fist). Besides, when action is too fast or hands occlude each other, the positions of hands are difficult to locate. Therefore, in HGR, it is better to avoid the dependence on the skeleton information offered by Kinect.

2.2 Classification

Classification can be integrated with feature extraction such as the boosting method including a combination of weak detectors. Other approaches include a distinct translation step into feature space and subsequent classification. The extracted features are then provided to classifiers, such as generic support vector machines and highly customized shape classifiers [32]. Hand gesture classification approaches are mainly machine learning based, in which mappings between feature sets and gestures are carried out by machine learning methods [33–36]. Fuzzy min–max NNs [37], adaptive neuro-fuzzy inference system networks [38], and hyper rectangular composite NNs [39] are used for hand shape classification. 3D Hopfield NN [40] is applied for sign classification. HMM-based classification, which was widely used in continuous speech recognition, has dominated HGR after the mid-1990s. Wang et al. [41] and Bauer et al. [42], respectively, implement 2D motion model and perform recognition with HMM. In the works [43, 44], HMM is improved to adapt to various situations.

Yet for the whole recognizing system, its performance does not only depend on the inherent properties of the object to be recognized and the design of classifier, but also on the size and quality of the training data. Traditional statistics is the science of asymptotic theory; the performance of various methods based on it can be guaranteed in theory only when the number of samples inclines toward infinity, for instance the uniformity of estimation, unbiased estimation, as well as the bound of the estimate of variance, all embodiment of this asymptotic theory. Therefore, large numbers of training data are required of the statistical model to get fairly satisfactory recognition performance. In the case of insufficient training data being available, complex models that have extensive parameters to learn are very likely to encounter the over fitting problem, such as HMM, AdaBoost, and decision trees methods [45, 46]. One disadvantage of collecting large numbers of gestures data is that the work load is heavy, especially in the insertion of new gesture. To facilitate the adaptation to new gestures, Kaggle is organizing a one-shot-learning challenge for gesture recognition, and some recent works exploiting one-shot learning [47–49] show interesting results. Focusing on the CHALEARN gesture challenge, this paper designs effective HGR in this challenging condition.

3 Two-layered hand gesture recognition

There are significant differences between postures and gestures although they are frequently considered as the same. A posture is a static pose, such as making a palm posture and holding it in a certain position, while a gesture is a dynamic process consisting of a sequence of the changing postures over a short duration. Compared to postures, gestures contain much richer motion information, which is important for distinguishing different gestures, especially those ambiguous ones. The main challenge of gesture recognition lies in the understanding of the unique characters of gestures. Exploring and utilizing these characters in gesture recognition are crucial for achieving desired performance. Two crucial linguistic models of gestures are the phonological model drawn from the component concurrent character and the movement-hold model drawn from the sequential organization character. The component concurrent character indicates that complementary components, namely motion, location and shape components, simultaneously characterize a unique gesture. Therefore, an ideal gesture recognition method should have the ability of capturing, representing, and recognizing these simultaneous components.

Based on the key cues in the gesture meaning expression and motivated by the above observation, this paper proposes a novel two-layered HGR architecture as shown in Fig. 1, which has two layers, namely the motion component classifier and the location component classifier.

In the proposed two-layered HGR, each layer analyzes its corresponding component. The output of the first layer limits the possible classification in the second layer. These classifiers complement each other for the final gesture classification. Once the motion component analysis and classification at the first layer is accomplished, the original

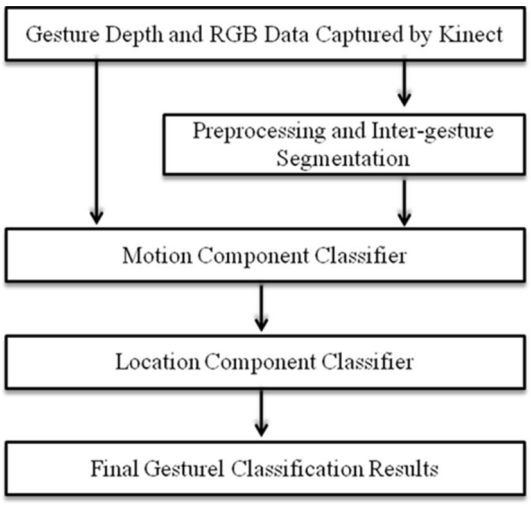


Fig. 1 Proposed two-layered hand gesture recognition

complete gesture vocabulary is divided into possible and impossible candidates' lists. The possible candidates' list is then input to the second layer for the location component analysis and classification. In the second layer, the best match gesture is output as the final gesture recognition result. These two layers analyze the gesture components, respectively, and weakly coupled in HGR, which ensures the effectiveness of their complementarily.

3.1 Motion component analysis and classification

In the hand gesture sequence, each frame has the relevant movement with respect to its adjacent frame and initial position. These movements and their statistical information are valuable in the analysis of the gesture sequence, which is processed and analyzed in the first layer. The principal motion [50] is improved in the process of analyzing and classifying the gesture motion component. The neighboring block overlapping is proposed in the partitioning of bins. Furthermore, the RGB data and depth information are concatenated as input. By excluding the aliened gesture candidates, a list of possible candidates is then forwarded to the second layer.

3.1.1 Principal motion method

In [50], Escalante and Guyon use a set of histograms of "motion energy" information to represent a gesture sequence and implement a reconstruction approach to HGR based on principal components analysis (PCA). For an N length gesture sequence, "motion energy" is calculated by subtracting consecutive frames (from frame 1 to $N-1$). Thus, the gesture with N frames is associated to $N-1$ "motion energy" images. Then, a grid of equally spaced N_b bins, as shown in Fig. 3c, is defined over the "motion energy" image. Upon averaging motion energy in each of the cells of the grid, an N_b bins' histogram for each difference image is obtained. Accordingly, an N frame gesture is represented by a matrix Y of dimensions $(N-1) \times N_b$. Having represented each training gesture sequence in the gesture vocabulary of size K with matrices Y_k, $k \in \{1, \ldots, K\}$, PCA is applied to each of these matrices Y_k. Then, the top c eigenvalues, W, and corresponding eigenvectors, U for each training gesture, i.e., $(W, U)_{\{1,\ldots,K\}}$ are stored.

In the recognition process, each test gesture is processed similarly as training gestures, and it is represented by a matrix of S. Then, S is projected into each of the K-spaces induced by $(W,U)_{\{1,\ldots,K\}}$ back. Assuming $R_1, \ldots, R_K$ denote the reconstructions of S according to the K PCA models, respectively, the reconstruction error for each R_k is measured as follows:

$$\varepsilon(k) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}(R_k(i,j) - S(i,j))^2},$$

where n and m are the number of rows and columns of S, respectively. Finally, the gesture is assigned to the PCA model that obtained the lowest reconstruction error, that is: $\text{argmin}_k\ (\varepsilon(k))$.

3.1.2 Improved principle motion and classification

Firstly, in the representation of motion, an overlapping neighborhood is adopted while computing the gird bins' values. Gestures are usually performed with significant deformation as shown in Fig. 3. In [50], motion representation is based on bins, which are strictly limited in position. Each bin is analyzed independently, having no consideration on the space interdependency among the neighboring bins. Actually, this interdependency is important to the effective expressing motion component of gesture, especially in the condition of gesture deformation. Accordingly, for each bin, the current 20×20 region, an overlapping neighborhood, is proposed which includes all 3×3 equally spaced neighbor bins in an 60×60 square region. The averaged motion energy in the square region is assigned to the current bin's value to provide more tolerance to subtle differences in large amplitude gestures, as shown in Fig. 2.

Secondly, principle motion is applied on both the RGB and depth data. The RGB image is transformed into gray image before subtracting consecutive frames. For each frame in the reference gesture sequence, two 192 bins' histograms are obtained with depth difference image and gray difference image, respectively.

For each reference gesture, the top 10 eigenvalues, $W_{(1\times10)}$, and corresponding eigenvectors, $U_{(192\times10)}$, are

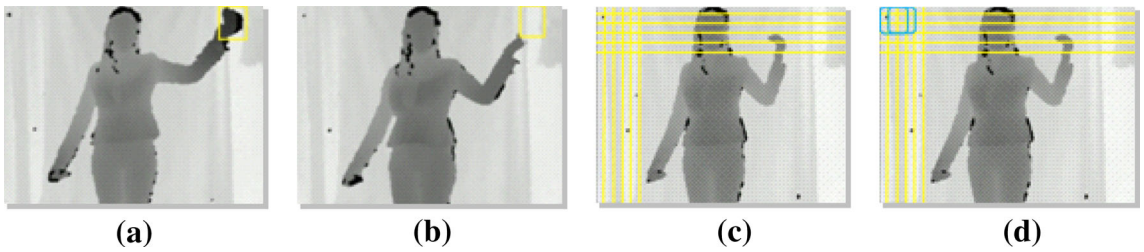


Fig. 2 Same people performing the same gesture with significant deformation (**a**, **b**), clipped form the development data CGD 2011. **c** Is the grid of equally spaced bins in principle motion. **d** Is a description of the overlapping neighborhood that includes all 3 × 3 equally spaced neighbor bins

stored as the reference gesture model in this paper. The final K reconstruction errors of test gesture based on the reference gesture models are obtained by multiplying the reconstruction error of depth data and that of the RGB data accordingly. Then, these K reconstruction errors are assigned as input for the division of the gesture vocabulary of size K into two categories, possible candidates and impossible candidates, for matching the test gesture. This is done using the K-means algorithm, and then, the possible candidates list is forwarded to the second layer. The recognition process is illustrated in Fig. 3.

We validate the performance of the proposed improved principal motion method on the first 20 development batches of CGD 2011. Using the provided code [50, 27] as baseline, in the case of principal motion, normalized Levenshtein distance [51] is 44.92 %, and in the case of improved principal motion, a better performance, a distance 38.66 % is achieved. The Levenshtein distance is also known as "edit distance," and this error metrics is in accordance with the Leaderboard in CHALEARN gesture challenge [21]. The first layer classifier can filter gestures with high accuracy and speed.

3.2 Location component analysis

Considering the scale and rotation variance of human's gestures in daily life, the SIFT and BOW model is regarded as one of most promising methods in one-shot-learning gesture recognition [52], while there are two major disadvantages that restrict the accuracy of recognition seriously. On the one hand, the SIFT algorithm is difficult to perform the temporal segmentation and gesture recognition concurrently, so pre-segmentation is needed to segment multiple gestures sequence into single gestures [51]. On the other hand, the common BOW algorithm generally puts all descriptors of the gesture together neglecting the significance of time order. Our proposed methodology aims at solving those two problems by building spatial pyramids [53] for DTW and temporal pyramids [54] for histograms of BOW model.

A normal method of using SIFT + BOW for gesture recognition is shown in Fig. 4. In the training process, the normal method transforms a single gesture video to a unique histogram by obtaining key points from RGB video, extracting descriptors of each key point from RGB and depth video, clustering descriptors to a codebook, and getting histogram to uniquely represent the single gesture video. In the testing process, since a test video may contain several single gestures, DTW + Viterbi is used to divide the test video into several single ones. And after getting the histograms, KNN is used for final classification. Our proposed method builds spatial pyramids for DTW when dividing multiple gestures video into single ones (marked red in Fig. 4) and builds temporal pyramids in both training and testing process. By making more use of spatial and temporal information than normal SIFT + BOW, the proposed method can earn a better performance.

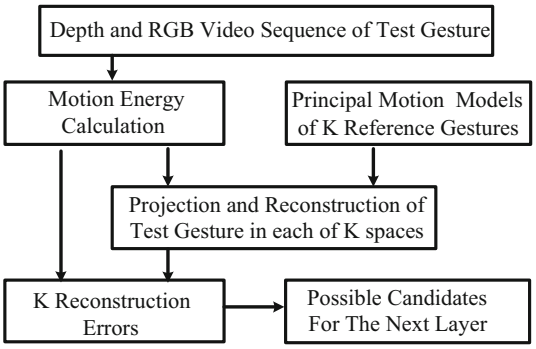


Fig. 3 Motion component analysis and classification

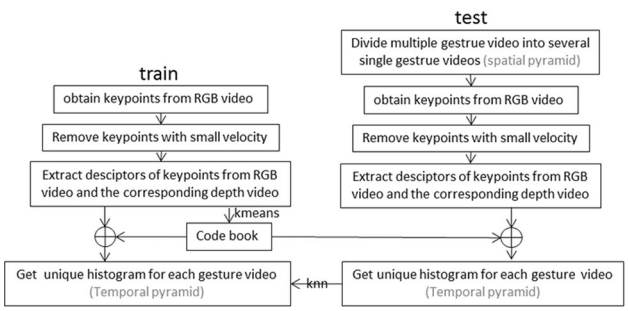


Fig. 4 General process of gesture location classification

3.2.1 Spatial pyramid

The gesture recognition algorithm can only recognize single gestures, so multiple gestures video should firstly be cut into several small sequences, i.e., linguistic units. The method of cutting multiple gestures is DTW + Viterbi algorithm. DTW is a dynamic programming algorithm, which can calculate the similarity of two time sequences. DTW inputs two time sequences and outputs a DTW distance. Lower DTW distance means higher similarity of two gesture videos. The normal DTW usually resizes a gesture video from $W \times H \times F$ sequence (namely Width * Height * Frames) to $3 \times 3 \times F$ sequence and then formats to a $9 \times F$ sequence.

On the one head, resizing frame of a gesture video to 3×3 matrix is too rough to contain all gesture features. On the other head, resizing it to 10×10 matrix or larger is so sensitive that a small bias of human gesture (which should be recognized as the same gesture) will lead to an obviously different matrixes. To contain more information without making it too sensitive, the proposed method adds spatial pyramid when using DTW. Each frame in the video is resized to 3×3 and 7×7 matrix, as shown in Fig. 5b. Then, transforming 3×3 and 7×7 matrix into a 58 dimension vector, the whole video will be transformed into a $58 \times F$ vector sequence. Finally, the Euclidean distance of the vectors is calculated. As shown in Fig. 5a, each block represents a Euclidean distance of two 58 dimension vectors. The similarity of two gesture videos can be expressed by cumulative distance, which accumulates local Euclidean distance from the first frame to the last frame.

The cumulative direction of DTW in each step is restricted as demonstrated in the following two formulas:

$$\gamma(i,j) = d(q_i, c_j) + \min\{(\gamma(i-1,j-2), \gamma(i-1,j-1), \gamma(i-1,j))\} \quad (1)$$

$$\gamma(i,j) = d(q_i, c_j) + \min\{(\gamma(i-1,j-3), \gamma(i-1,j-2), \gamma(i-1,j-1), \gamma(i-1,j))\} \quad (2)$$

where q and c are DTW vector sequences of videos, and $\gamma(i, j)$ is the cumulative distance at (i,j). Formula 2 is suitable for the situation that the gesture velocity of q is similar to the gesture velocity of c. And Formula (3) is suitable for the situation that the gesture velocity of q is fairly faster the gesture velocity of c.

To divide test-multiple-gestures video, Viterbi is used together with DTW. The test-multiple-gestures can be considered as a state sequence, the train-single-gesture is a state, and DTW distance represents the possibility of the state. So Viterbi algorithm can be used to calculate the implicit state sequence. The general idea is finding the most possible combination of train-single-gesture videos, which has the minimum DTW cumulative distance with the test-multiple-gestures video. Specifically, in Fig. 6 $58 \times F$ dimension test-multiple-gestures sequence serves as x-axis, and 10 train-single-gesture sequence connecting together as y-axis. All combination of train-single-gesture sequences and their DTW distance with the test-multiple-gestures sequence will be calculated. And finally, the optimal DTW path will be found (green line in Fig. 6), and then, the divide point of test-multiple-gestures video will be found as well (red arrow in Fig. 6). There are in fact many tricks to accelerate those processes rather than repeatedly calculate all combinations, and the implementation of normal DTW + Viterbi can be found in [52].

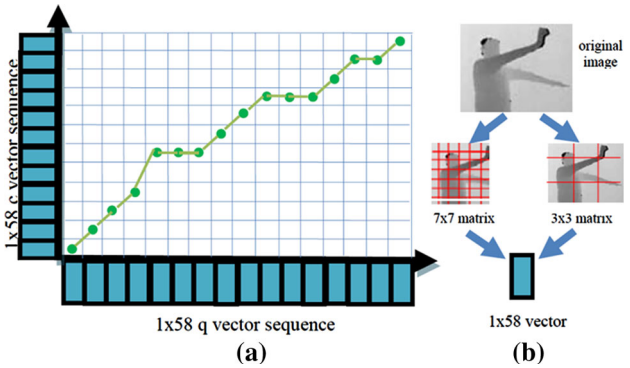


Fig. 5 Spatial pyramid building in DTW. **a** Calculate Euclidean distances of two-vector sequence by DTW. **b** Transform one frame into a 1×58 vector

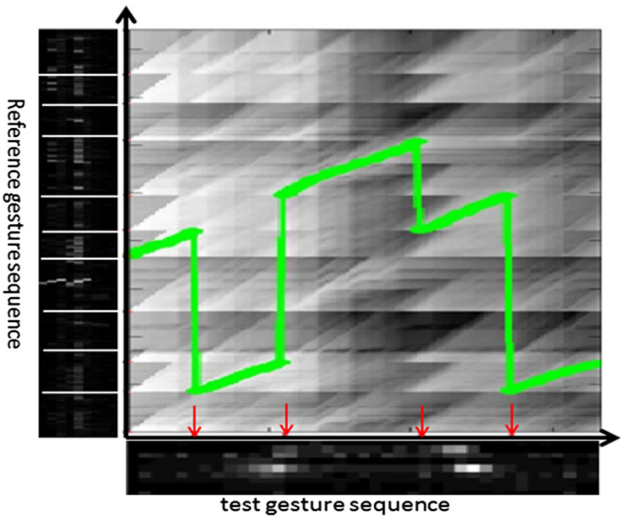


Fig. 6 DTW with Viterbi to divide test-multiple-gestures video

3.2.2 Temporal pyramid

After segmenting multiple-gesture video into single ones, the SIFT and BOW model is used to extract unique features from every single-gesture video.

There are three steps in SIFT model. Step one is detecting key points with Gaussian pyramid, noting that the key points should get from RGB frame rather than depth frame because of the strong impulse noise in depth frame. The step two is removing the key points with small velocity. Optical flow is adopted to compute the velocity of each key point, after that the key points with the velocity less than a fix threshold will be discarded. The third step is calculating the descriptors around the filtered key points so that the whole single gesture can be represented by several local descriptors. Each descriptor is a vector with 128×6 elements containing gradient and velocity feature in xy, xz, yz plane [52].

The descriptors extracted from SIFT model have different lengths. In order to compare the features of each gesture, the BOW model is applied. The process of BOW is as follows: K-means algorithm [55] is used to make the features transformed into a unified dimension. The algorithm clusters all descriptors into k centers to compose a visual codebook, and then, each descriptor can be described by the visual codebook. The number of centers can be calculated as follows:

$$k = \alpha \times n \tag{3}$$

where n is the number of descriptors, and the constant value α is 0.4. The codebook calculated by k-means is a $k \times 768$ matrix, and each center of the codebook is a 1×768 vector. And the descriptors are also 1×768 vectors. So the distance between centers and descriptors is Euclidean distance, which is easy to compute.

After that a histogram is adopted to statistics the descriptors. The histogram is a $k \times 1$ vector with statistical information, which can represent a single gesture uniquely. Then, KNN is used to match the histogram in test process with the histogram in training process to get the final result of gesture recognition.

According to the above description, the method of creating histogram has great influence on gesture recognition. The normal k-means mixes all key points (red points in Fig. 7) together, neglecting temporal relationship of the key points, so the normal k-means loses useful information about temporal order. To make full use of the temporal information, a temporal pyramid is built when creating histograms as shown in Fig. 7.

A single-gesture video is cut into two parts at the middle of the video. Each part has half number of the total frames. The part1 of the video has 3 key points, and the part2 has 2 key points. Put the key points in part1 into codebook, we will get a $k \times 1$ histogram for part1. Put the key points in part2 into codebook, we will get a $k \times 1$ histogram for part2. And put all $3 + 2$ key points into codebook, we will get a $k \times 1$ histogram for whole video. Our proposed histogram connects three $k \times 1$ histograms together. "Histogram for whole video" is the normal histogram extract by SIFT + BOW (base layer of pyramid); "Histogram for whole video" and "Histogram for whole video" are the temporal constraints for gesture video (upper layer pyramid). Combining those three histograms can effectively capture more useful information for gesture recognition.

3.3 From component level to final gesture classification

In the proposed HGR, the classifiers complement each other. Once the motion component analysis and classification at the first layer is accomplished, the original complete gesture vocabulary is divided into possible and impossible candidates' lists. The possible candidates' list is then input to the second layer for the location component analysis and classification. The best match gesture is output as the final gesture recognition result.

4 Experiments

In this section, extensive experimental results are presented to evaluate the proposed multi-layered HGR performance. All the experiments are performed in MATLAB 7.12.0 on Dell with Duo CPU E8400. ChaLearn Gesture Dataset (CGD2011) is used in the experiments, which is designed for one-shot learning. CGD2011 is the largest gestures dataset recorded with Kinect [19], which consists of 50,000 gestures (grouped in 500 batches, each batch including 47 sequences and each sequence containing of 1–5 gestures drawn from one of 30 small gesture vocabularies of 8–15 gestures), with frame size 240×320, 10 frames/second, recorded by 20 different users. In our experiments, Levenshtein distance is used to evaluate the HGR performance, which is also used in CHALEARN gesture challenge. Levenshtein distance is the minimum number of edit operations that have to perform from one sequence to another, and the Levenshtein distance is also known as "edit distance" [56].

4.1 Proposed location component HGR evaluation

There are three experiments in this part. The effects of applying spatial pyramid and temporal pyramid are evaluated, respectively. Experiment C compares the proposed method with other state-of-the-art methods.

4.1.1 Spatial pyramid

The spatial pyramid can be built based on either RGB data or depth data. So before building the pyramid, an experiment on comparing RGB data and depth data is conducted.

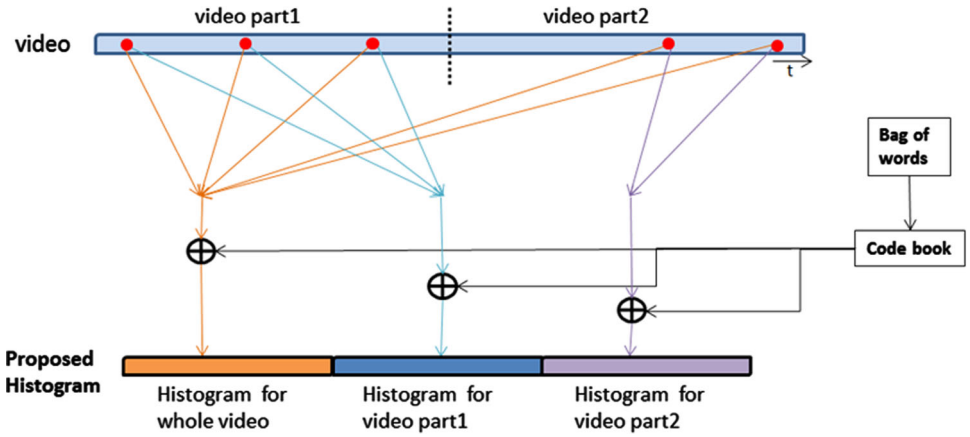


Fig. 7 Process of building temporal pyramid

As illustrated in Sect. 2, each vector in the DTW vector sequence contains 58 × 1 elements. To test the effect of RGB data in DTW, the RGB image is resized to a 3 × 3 matrix. Then, the 3 × 3 matrix is transformed into a 9 × 1 vector and replace the 58 × 1 vector. The process of testing depth data is the same as the process of testing RGB data. To evaluate the performance of RGB data and depth data, the number of the videos after pre-segment is counted. DTW and Viterbi algorithm segment multiple gestures video into single gesture. If the total number of videos is less than 100 after being processed by DTW and Viterbi algorithm, it means that there are some multiple gestures, which were not cut by DTW and Viterbi algorithm. And if the total number of videos is more than 100, it means that some single gestures were wrongly cut by DTW and Viterbi algorithm. Figure 8 shows the effect of adding temporal pyramid in DTW.

The result in Fig. 8 shows that the adoption of temporal pyramid decreases the variance of the result, which makes the pre-segment more robust and precise. It is demonstrated that the result of using depth data for DTW and Viterbi algorithm is better than the result of using RGB data. In fact, any attempt to combine depth data with RGB data cannot have a better performance than using depth data alone. For instance, the final 27 dataset is difficult for gesture recognition, and the number of videos after pre-segment for final 27 is only 81, as shown in Fig. 8. The other experiment is about the restriction of cumulative path in DTW. The comparison between the result of adopting formula (1) and formula (2) is shown in Fig. 9.

If the number of videos after pre-segment is more than 100, it means that some single gestures are not recognized correctly as a series of multiple gestures, which reduces the final accuracy of gesture recognition drastically. So although the average number of pre-segment videos for formula (2) is closer to 100 than formula (1) as shown in Fig. 9, formula (1) can get a lower error rate of gesture recognition as Table 1 shows. Figure 9 together with Table 1 demonstrates that

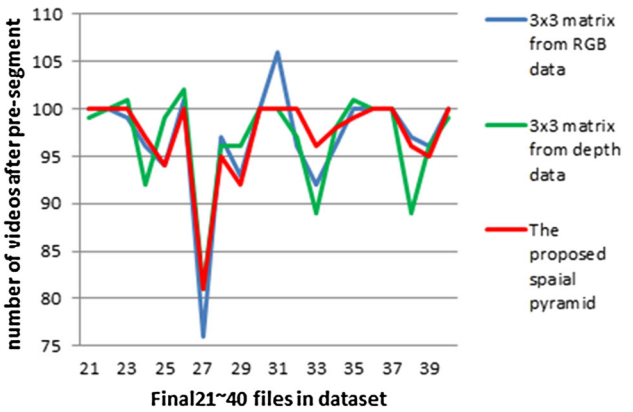


Fig. 8 Comparison between the normal DTW (with 3 × 3 matrix)

formula (1) is suitable for normal situation that the velocity of train gesture is similar to the one of test gesture. And formula (2) beats formula (1) obviously for the situation that the velocity of train gesture is fairly faster than the one of test gesture, such as final 7 dataset.

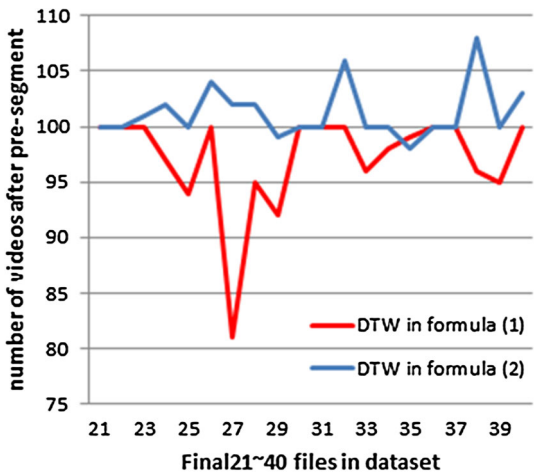


Fig. 9 Comparison between adopting and the proposed method (with temporal pyramid) formulas (1) and (2)

4.1.2 Temporal pyramid

An experiment comparing the normal histogram and histogram with temporal pyramid is performed. The evaluation criterion of the final result of gesture recognition is LD score. LD score is a measurement of the error rate. Lower LD score means higher accuracy of gesture recognition. Table 2 compares the LD score between normal histogram and long histogram with temporal pyramid. K-means is random function, so the LD score changes in a small scale even if nothing changes in the program. The experiment is conducted for three times to ensure the reliability of the final result.

4.1.3 Compared with other methods

The proposed method is compared with other popular methods [57, 58] in Fig. 7. The independent variable of x-axis is the α in formula (3). The variable α determines the size of the histograms and has an important impact on the final result.

Figure 10 shows that the proposed method can obtain high performance than other state-of-the-art methods. In addition, an experiment comparing RGB data with RGB-D data is conducted. It demonstrates that for the proposed

Table 1 Levenshtein distance for formula (1) and formula (2) on first 20 development batches of CGD2011

Dataset	Formula (%)	
	Formula (1)	Formula (2)
Final 1	9	10
Final 2	1	1
Final 3	9	12
Final 4	24	27
Final 5	13	**12**
Final 6	12	15
Final 7	33	**29**
Final 8	33	34
Final 9	15	16
Final 10	10	12
Final 11	2	2
Final 12	15	22
Final 13	14	**13**
Final 14	23	24
Final 15	2	2
Final 16	0	0
Final 17	0	0
Final 18	20	28
Final 19	21	**20**
Final 20	7	8
Average	13.3	14.4

Table 2 Comparison between normal histogram and long histogram with temporal pyramid in LD score

Batch	Normal	Temporal pyramid
1	0.134	0.128
2	0.126	0.121
3	0.130	0.123
Average	0.130	0.124

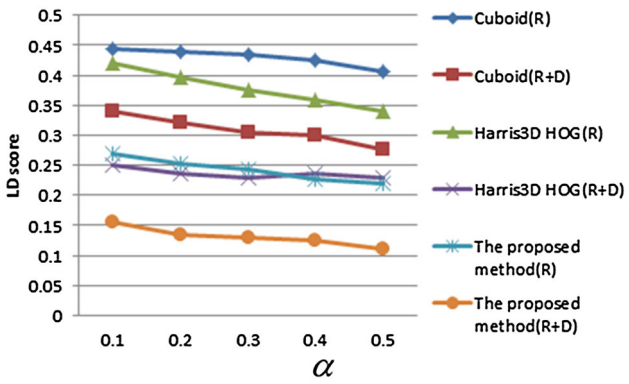


Fig. 10 Comparison with other methods by adopting different α

Table 3 Recognition performance of using the second layer, first two layers, and three layers on first 20 development batches of CGD2011 [19] (TeLev is the average Levenshtein distance)

Batch number	First two layers for recognition	
	TeLev (%)	Recognize time per gesture (s)
1	8.70	2.90
2	5.21	6.60
3	8.75	2.49
4	23.67	1.60
5	12.62	2.55
6	11.94	2.19
7	14.51	1.20
8	0.00	1.64
9	12.44	2.00
10	9.13	1.74
11	1.36	3.48
12	11.06	1.50
13	12.93	0.70
14	10.13	0.40
15	4.21	0.59
16	6.27	3.71
17	9.55	4.60
18	22.21	0.31
19	17.32	2.34
20	8.23	0.90
Average	**12.90**	**2.22**

Table 4 Performance comparison on the 20 development data batches (TeLen is the average error made on the number of gestures)

Methods	Extend MHI [47]	Manifold LSR [48]	3DHOF + GHOG [49]	Sparse coding [49]	Temporal Bayesian [60]	Motion history [59]	Proposed
TeLev (%)	26.00	28.73	43.32	25.11	24.09	31.25	12.90
TeLen (%)	#	6.24	5.02	5.02	#	18.01	5.76

method, 3D sensors that capture RGB data and depth data have an obvious advantage over normal sensors in gesture recognition.

4.2 Proposed two-layered HGR evaluation

Table 3 shows the HGR performance on the first 20 development batches. For the first layer, the average Levenshtein distance is 24.02 % with total 1.03 s/gesture. For the first two layers, the average Levenshtein distance is 12.79 % with total 2.73 s/gesture.

For the comparison on the first 20 batches of CGD2011, the proposed multilayered HGR is further compared with Lui [48]: a nonlinear regression framework on manifolds (Manifold LSR), Wu et al. [47]: extended-motion-history-image and maximum correlation coefficient (extended MHI), and Mahbub et al. [59]: the motion history-based silhouettes and Euclidean distance-based classifiers (motion history) on the first 20 development data batches, as listed in Table 4.

The first layer is fast and robust. The correct gesture can be mostly identified as the possible candidate within approximately 80 fps (frames per second). For the second layer, the average Levenshtein distance is 14.4 % and its computing time is 10 times than that of the first layer. For the two-layer HGR, the average Levenshtein distance is 19.62 % and 15 fps is achieved (faster than 10 fps in CGD2011).

5 Conclusion

This paper proposes a novel HGR system using Kinect. The proposed recognition architecture and the choice of algorithms take into account both the simultaneous components and sequential organization of gestures. The analysis and classification of hand gesture basic components: motion, location and hand shape, integrate into a multi-layered framework, according to the consideration on accuracy and computation complexity. High recognition accuracy and real-time performance are achieved. In the future, research work may focus on more advanced appearance-based descriptor, which could better assist the system in discriminating static gestures and explores the adoption of more complex classification model with enlarged training set of synthetically generated dynamic multi-stream samples.

Acknowledgments We would like to acknowledge the editors and reviewers, whose valuable comments greatly improved the manuscript. This work was supported in part by the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant Nos. 61572155 and 61272386.

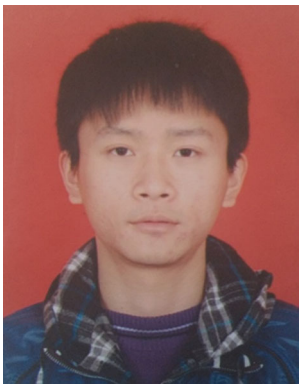
References

1. Kendon, A.: Visible Action as Utterance. Cambridge University Press, Cambridge (2004)
2. Mitra, S., Acharya, T.: Gesture recognition: a survey. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **37**(3), 311–324 (2007)
3. Fang, G., Gao, W., Zhao, D.: Large-vocabulary continuous sign language recognition based on transition-movement models. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **37**(1), 1–9 (2007)
4. Baklouti, M., Monacelli, E., Guitteny, V., Couvet, S.: Intelligent assistive exoskeleton with vision based interface. In: Proceedings of the 6th International Conference on Smart Homes and Health Telematics, vol. 5120, pp. 123–135 (2008)
5. Nickel, K., Stiefelhagen, R.: Visual recognition of pointing gestures for human-robot interaction. Image Vis. Comput. **25**(12), 1875–1884 (2007)
6. Wu, Y., Huang, T.S.: Hand modeling analysis and recognition for vision based human computer interaction. IEEE Signal Process. Mag. Spec. Issue Immers. Interact. Technol. **18**(3), 51–60 (2001)
7. Corradini, A.: Real-time gesture recognition by means of hybrid recognizers. In: International Workshop on Gesture and Sign Languages in Human–Computer Interaction, vol. 2298, pp. 34–46 (2001)
8. Ong, S.C., Ranganath, S.: Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 873–891 (2005)
9. Aran, O., Keskin, C., Akarun, L.: Computer applications for disabled people and sign language tutoring. In: Proceedings of the Fifth GAP Engineering Congress, pp. 26–28 (2006)
10. Cooper, H., Holt, B. Bowden, R.: Sign language recognition. In: Visual Analysis of Humans. Springer, London, pp. 539–562 (2011)
11. Triesch, J., Malsburg, C.: Robotic gesture recognition by cue combination. In: Gesture and Sign Language in Human–Computer Interaction. Lecture Notes in Computer Science. Springer, Berlin, pp. 233–244 (1998)
12. Hong, S., Setiawan, N.A., Lee, C.: Real-time vision based gesture recognition for human robot interaction. In: Proceedings of

International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, vol. 4692, p. 493 (2007)
13. Grzeszcuk, R., Bradski, G., Chu, M.H., Bouguet, J.Y.: Stereo based gesture recognition invariant to 3d pose and lighting. In: Proceedings of CVPR, pp. 826–833 (2000)
14. Fujimura, K., Liu, X.: Sign recognition using depth image streams. In: Proceedings of FGR, Southampton, UK, pp. 381–386 (2006)
15. Hadfield, S., Bowden, R.: Generalised pose estimation using depth. In: Proceedings of ECCV International Workshop: Sign, Gesture, Activity, Heraklion, Crete (2010)
16. Ershaed, H., Al-Alali, I., Khasawneh, N., Fraiwan, M.: An Arabic sign language computer interface using the Xbox Kinect. In: Annual Undergraduate Research Conference on Applied Computing, Dubai, UAE (2011)
17. Hong, R., Wang, M., Gao, Y., Tao, D., Li, X., Wu, X.: Image annotation by multiple-instance learning with discriminative feature mapping and selection. IEEE Trans. Cybern. **44**(5), 669–680 (2014)
18. Suma, E., Lange, B., Rizzo, A., Krum, D., Bolas, M.: FAAST: The flexible action and articulated skeleton toolkit. In: IEEE Virtual Reality Conference, pp. 247–248 (2011)
19. Guyon, I., Athitsos, V., Jangyodsuk, P., Hamner, B., Escalante, H.J.: Chalearn gesture challenge: Design and first results. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–6 (2012)
20. Zhou, H., Huang, T.: Tracking articulated hand motion with Eigen dynamics analysis. In: Proceedings of International Conference on Computer Vision, vol. 2, pp. 1102–1109 (2003)
21. Wu, Y., Lin, J., Huang, T.: Capturing natural hand articulation. In: IEEE International Conference on Computer Vision, pp. 426–432 (2001)
22. Dardas, N.H.A.Q.: Real-time hand gesture detection and recognition for human computer interaction. Ottawa-Carleton Institute for Electrical and Computer Engineering, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario (2012)
23. Kadir, T., Bowden, R., Ong, E.J., Zisserman, A.: Minimal training, large lexicon, unconstrained sign language recognition. In: Proceedings of BMVC, Kingston, UK, pp. 939–948 (2004)
24. Zhang, L.G., Chen, Y., Fang, G., Chen, X., Gao, W.: A vision-based sign language recognition system using tied-mixture density HMM. In: Proceedings of International Conference on Multimodal interfaces, State College, PA, USA, pp. 198–204 (2004)
25. Awad, G., Han, J., Sutherland, A.: A unified system for segmentation and tracking of face and hands in sign language recognition. In: Proceedings of ICPR, Hong Kong, China, pp. 239–242 (2006)
26. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Eng. Appl. Artif. Intell. **22**, 1141–1158 (2009)
27. Maung, T.H.H.: Real-time hand tracking and gesture recognition system using neural networks. World Acad. Sci. Eng. Technol. **50**, 466–470 (2009)
28. Maraqa, M., Abu-Zaiter, R.: Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In: International Conference on Applications of Digital Information and Web Technologies, pp. 478–481 (2008)
29. Akyol, S., Alvarado, P.: Finding relevant image content for mobile sign language recognition. In: International Conference on Signal Processing, Pattern Recognition and Application, Rhodes, Greece, pp. 48–52 (2001)
30. Wong, S.F., Kim, T.K., Cipolla, R.: Learning motion categories using both semantic and structural information. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–6 (2007)
31. Zahedi, M., Keysers, D. Ney, H.: Appearance-based recognition of words in American sign language. In: Pattern Recognition and Image Analysis. Springer, Heidelberg, pp. 511–519 (2005)
32. Yin, X., Zhu, X.: Hand posture recognition in gesture-based human–robot interaction. In: IEEE Conference on Industrial Electronics and Applications, pp. 1–6 (2006)
33. Chen, B.W., He, X., Ji, W., Rho, S., Kung, S.Y.: Support vector analysis of large-scale data based on kernels with iteratively increasing order. J. Supercomput., 1–15 (2015)
34. Chen, B.W., Wang, J.C., Wang, J.F.: A novel video summarization based on mining the story-structure and semantic relations among concept entities. IEEE Trans. Multimedia **11**(2), 295–312 (2009)
35. Chen, B.W., Chen, C.Y., Wang, J.F.: Smart homecare surveillance system: behavior identification based on state transition support vector machines and sound directivity pattern analysis. IEEE Trans. Syst. Man Cybern. Syst. **43**(6), 1279–1289 (2013)
36. Jiang, F., Wu, S., Yang, G., Zhao, D., Kung, S.Y.: Viewpoint-independent hand gesture recognition with Kinect. SIViP **8**(1), 163–172 (2014)
37. Simpson, P.: Fuzzy min-max neural networks—part 1: classification. IEEE Trans. Neural Netw. **3**, 776–786 (1992)
38. Al-Jarrah, O., Halawani, A.: Recognition of gestures in Arabic sign language using neuro-fuzzy systems. Artif. Intell. **133**, 117–138 (2001)
39. Su, M.C.: A fuzzy rule-based approach to spatio-temporal hand gesture recognition. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **30**(2), 276–281 (2000)
40. Huang, C.L., Huang, W.Y.: Sign language recognition using model-based tracking and a 3D Hopfield neural network. Mach. Vis. Appl. **10**(6), 292–307 (1998)
41. Wang, C., Gao, W., Shan, S.: An approach based on phonemes to large vocabulary Chinese sign language recognition. In: Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 393–398 (2002)
42. Bauer, B., Kraiss, K.F.: Video-based sign recognition using self-organizing subunits. In: Proceedings of International Conference on Pattern Recognition, vol. 2, pp. 434–437 (2002)
43. Tanibata, N., Shimada, N., Shirai, Y.: Extraction of hand features for recognition of sign language words. In: Proceedings of International Conference on Vision Interface, pp. 391–398 (2002)
44. Hong, R., Wang, M., Li, G., Nie, L., Zha, Z.J., Chua, T.S.: Multimedia question answering. IEEE Multimedia **19**(4), 72–78 (2012)
45. Jiang, F., Gao, W., Yao, H., Zhao, D., Chen, X.: Synthetic data generation technique in Signer-independent sign language recognition. Pattern Recogn. Lett. **30**(5), 513–524 (2009)
46. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
47. Wu, D., Zhu, F., Shao, L.: One shot learning gesture recognition from rgbd images. In: CVPR Workshop on Gesture Recognition, pp. 7–12 (2012)
48. Lui, Y.M.: A least squares regression framework on manifolds and its application to gesture recognition. In: CVPR Workshop on Gesture Recognition, pp. 13–18 (2012)
49. Fanello, S. R., Gori, I., Metta, G., Odone, F.: One-shot learning for real-time action recognition. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 31–40. Madeira, Portugal (2013)
50. Escalante, H.J., Guyon, I., Athitsos, V., Jangyodsuk, P., Wan, J.: Principal motion components for one-shot gesture recognition. Pattern Anal. Appl. 1–16 (2015). doi:10.1007/s10044-015-0481-3
51. Keogh, E., Ratanamahatana, C.A.: exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)

52. Wan, J., Ruan, Q., Li, W., An, G., Zhao, R.: 3D SMoSIFT: three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos. J. Electron. Imaging **23**(2), 023017 (2014)
53. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 2, pp. 2169–2178 (2006)
54. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: IEEE International Conference on Computer Vision, 2005, vol. 2, pp. 1458–1465 (2005)
55. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
56. Levenshtein, V.I.: February. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, no. 8, pp. 707–710 (1966)
57. Harris, C., Stephens, M.: A combined corner and edge detector. In Alvey Vision Conference, vol. 15, pp. 50–54 (1988)
58. Dollár, P., Rabaud, V., Gottrell, G.: Behavior recognition via sparse spatio-temporal features. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72 (2005)
59. Mahbub, U., Roy, T., Rahman, M.S, Imtiaz, H.: One-shot-learning gesture recognition using motion history based gesture silhouettes. In: Proceedings of the International Conference on Industrial Application Engineering, pp. 186–193 (2013)
60. Malgireddy, M.R., Inwogu, I., Govindaraju, V.: A temporal Bayesian model for classifying, detecting and localizing activities in video sequences. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 43–48 (2012)

Feng Jiang received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2001, 2003, and 2008, respectively. He is now an Associated Professor in the Department of Computer Science, HIT and a visiting scholar in the School of Electrical Engineering, Princeton University. His research interests include computer vision, pattern recognition, and image and video processing.



Jie Ren received the B.S. degrees from Department of Mathematics and School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin, China, in 2014. He is now working toward the Master degree at School of Computer Science and Technology, HIT.

Changhoon Lee received his Ph.D. degree in Graduate School of Information Management and Security (GSIMS) from Korea University, Korea. In 2008, he was a research professor at the Center for Information Security Technologies in Korea University. In 2009–2011, he was a professor in the School of Computer Engineering in Hanshin University. He is now a professor at the Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Korea. He has been serving not only as chairs, program committee, or organizing committee chair for many international conferences and workshops but also as a (guest) editor for international journals by some publishers. His research interests include information security, cryptography, digital forensics, smart grid security, and computer theory. He is currently a member of the IEEE, IEEE Computer Society, IEEE Communications, IACR, KIISC, KDFS, KIPS, KITCS, KMMS, KONI, and KIIT societies.

Wuzhen Shi received the B.S. degrees from Department of Mathematics and School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin, China, in 2015. He is now working toward the Master degree at School of Computer Science and Technology, HIT. His research interests include computer vision, pattern recognition, and image and video processing.

Shaohui Liu received the B.S., M.S., and Ph.D. degrees in Computer Science from Harbin Institute of Technology (HIT), Harbin, China, in 2000, 2002, and 2007, respectively. He is now an Associated Professor in the Department of Computer Science, HIT, and his research interests include data compression, pattern recognition, and image and video processing.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in Computer Science from Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively. He is now a Professor in the Department of Computer Science, HIT. He has published over 200 technical articles in refereed journals and conference proceedings in the areas of image and video coding, video processing, video streaming and transmission, and pattern recognition.