Journal of Machine Learning Research 16 (2015) 227-254 Submitted 12/13; Revised 7/14; Published 2/15

Multi-layered Gesture Recognition with Kinect

Feng Jiang FJIANG@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Harbin 150001, China

Shengping Zhang S.ZHANG@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Weihai 264209, China

Shen Wu WU.SHEN.ELTSHAN@GMAIL.COM
Yang Gao LAMBYY.HIT@GMAIL.COM
Debin Zhao DBZHAO@HIT.EDU.CN
School of Computer Science and Technology
Harbin Institute of Technology, Harbin 150001, China

Editors: Isabelle Guyon, Vassilis Athitsos, and Sergio Escalera

Abstract

This paper proposes a novel multi-layered gesture recognition method with Kinect. We explore the essential linguistic characters of gestures: the components concurrent character and the sequential organization character, in a multi-layered framework, which extracts features from both the segmented semantic units and the whole gesture sequence and then sequentially classifies the motion, location and shape components. In the first layer, an improved principle motion is applied to model the motion component. In the second layer, a particle-based descriptor and a weighted dynamic time warping are proposed for the location component classification. In the last layer, the spatial path warping is further proposed to classify the shape component represented by unclosed shape context. The proposed method can obtain relatively high performance for one-shot learning gesture recognition on the ChaLearn Gesture Dataset comprising more than 50, 000 gesture sequences recorded with Kinect.

Keywords: gesture recognition, Kinect, linguistic characters, multi-layered classification, principle motion, dynamic time warping

1. Introduction

Gestures, an unsaid body language, play very important roles in daily communication. They are considered as the most natural means of communication between humans and computers (Mitra and Acharya, 2007). For the purpose of improving humans' interaction with computers, considerable work has been undertaken on gesture recognition, which has wide applications including sign language recognition (Vogler and Metaxas, 1999; Cooper et al., 2012), socially assistive robotics (Baklouti et al., 2008), directional indication through pointing (Nickel and Stiefelhagen, 2007) and so on (Wachs et al., 2011).

Based on the devices used to capture gestures, gesture recognition can be roughly categorized into two groups: wearable sensor-based methods and optical camera-based methods. The representative device in the first group is the data glove (Fang et al., 2004), which is capable of exactly capturing the motion parameters of the user's hands and therefore can achieve high recognition performance. However, these devices affect the naturalness of the user interaction. In addition, they are also expensive, which restricts their practical applications (Cooper et al., 2011). Different from the wearable devices, the second group of devices are optical cameras, which record a set of images overtime to capture gesture movements in a distance. The gesture recognition methods based on these devices recognize gestures by analyzing visual information extracted from the captured images. That is why they are also called vision-based methods. Although optical cameras are easy to use and also inexpensive, the quality of the captured images is sensitive to lighting conditions and cluttered backgrounds, thus it is very difficult to detect and track the hands robustly, which largely affects the gesture recognition performance.

Recently, the Kinect developed by Microsoft was widely used in both industry and research communities (Shotton et al., 2011). It can capture both RGB and depth images of gestures. With depth information, it is not difficult to detect and track the user's body robustly even in noisy and cluttered backgrounds. Due to the appealing performance and also reasonable cost, it has been widely used in several vision tasks such as face tracking (Cai et al., 2010), hand tracking (Oikonomidis et al., 2011), human action recognition (Wang et al., 2012) and gesture recognition (Doliotis et al., 2011; Ren et al., 2013). For example, one of the earliest methods for gesture recognition using Kinect is proposed in Doliotis et al. (2011), which first detects the hands using scene depth information and then employs Dynamic Time Warping for recognizing gestures. Ren et al. (2013) extracts the static finger shape features from depth images and measures the dissimilarity between shape features for classification. Although, Kinect facilitates us to detect and track the hands, exact segmentation of finger shapes is still very challenging since the fingers are very small and form many complex articulations.

Although postures and gestures are frequently considered as being identical, there are significant differences (Corradini, 2002). 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 (Stokoe, 1960) and the movement-hold model drawn from the sequential organization character (Liddell and Johnson, 1989). 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. On the other hand, the movement phases, i.e., the transition phases, are defined as periods during which some components, such as the shape component, are in transition; while the holding phases are defined as periods during which all components are static. The sequential organization character characterizes a gesture as a sequential arrangement of movement phases and holding phases. Both the movement phases and the holding phases are defined as semantic units. Instead of taking the entire gesture sequence as input, the movement-hold model inspires us to segment a gesture sequence into sequential semantic units and then extract specific features from them. For example, for the frames in a holding phase, shape information is more discriminative for classifying different gestures.

It should be noted that the component concurrent character and the sequential organization character demonstrate the essences of gestures from spatial and temporal aspects, respectively. The former indicates which kinds of features should be extracted. The later implies that utilizing the cycle of movement and hold phases in a gesture sequence can accurately represent and model the gesture. Considering these two complementary characters together provides us a way to improve gesture recognition. Therefore, we developed a multi-layered classification framework for gesture recognition. The architecture of the proposed framework is shown in Figure 1, which contains three layers: the motion component classifier, the location component classifier, and the shape component classifier. Each of the three layers analyzes its corresponding component. The output of one layer limits the possible classification in the next layer and these classifiers complement each other for the final gesture classification. Such a multi-layered architecture assures achieving high recognition performance while being computationally inexpensive.

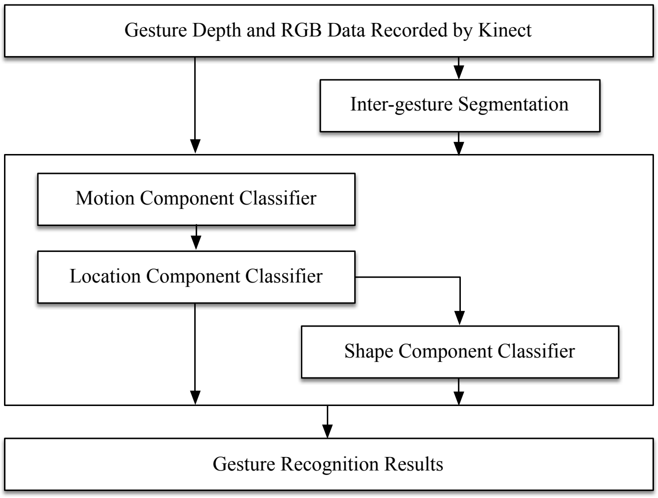
Gesture Depth and RGB Data Recorded by Kinect
Inter-gesture Segmentation
Motion Component Classifier
Location Component Classifier
Shape Component Classifier
Gesture Recognition Results

Figure 1: Multi-layered gesture recognition architecture.

The main contributions of this paper are summarized as follows:

- The phonological model (Stokoe, 1960) of gestures inspires us to propose a novel multi-layered gesture recognition framework, which sequentially classifies the motion, location and shape components and therefore achieves higher recognition accuracy while having low computational complexity.
- Inspired by the linguistic sequential organization of gestures (Liddell and Johnson, 1989), the matching process between two gesture sequences is divided into two steps: their semantic units are matched first, and then the frames inside the semantic units are further registered. A novel particle-based descriptor and a weighted dynamic time warping are proposed to classify the location component.

- The spatial path warping is proposed to classify the shape component represented by unclosed shape context, which is improved from the original shape context but the computation complexity is reduced from $O(n^3)$ to $O(n^2)$.

Our proposed method participated the one-shot learning ChaLearn gesture challenge and was top ranked (Guyon et al., 2013). The ChaLearn Gesture Dataset (CGD 2011) (Guyon et al., 2014) is designed for one-shot learning and comprises more than 50, 000 gesture sequences recorded with Kinect. The remainder of the paper is organized as follows. Related work is reviewed in Section 2. The detailed descriptions of the proposed method are presented in Section 3. Extensive experimental results are reported in Section 4. Section 5 concludes the paper.

2. Related Work

Vision based gesture recognition methods encompasses two main categories: three dimensional (3D) model based methods and appearance based methods. The former computes a geometrical representation using the joint angles of a 3D articulated structure recovered from a gesture sequence, which provides a rich description that permits a wide range of gestures. However, computing a 3D model has high computational complexity (Oikonomidis et al., 2011). In contrast, appearance based methods extract appearance features from a gesture sequence and then construct a classifier to recognize different gestures, which have been widely used in vision based gesture recognition (Dardas, 2012). The proposed multi-layered gesture recognition falls into the appearance based methods.

2.1 Feature Extraction and Classification

The well known features used for gesture recognition are color (Awad et al., 2006; Maraqa and Abu-Zaiter, 2008), shapes (Ramamoorthy et al., 2003; Ong and Bowden, 2004) and motion (Cutler and Turk, 1998; Mahbub et al., 2013). In early work, color information is widely used to segment the hands of a user. To simplify the color based segmentation, the user is required to wear single or differently colored gloves (Kadir et al., 2004; Zhang et al., 2004). The skin color models are also used (Stergiopoulou and Papamarkos, 2009; Maung, 2009) where a typical restriction is wearing of long sleeved clothes. When it is difficult to exploit color information to segment the hands from an image (Wan et al., 2012b), motion information extracted from two consecutive frames is used for gesture recognition. Agrawal and Chaudhuri (2003) explores the correspondences between patches in adjacent frames and uses 2D motion histogram to model the motion information. Shao and Ji (2009) computes optical flow from each frame and then uses different combinations of the magnitude and direction of optical flow to compute a motion histogram. Zahedi et al. (2005) combines skin color features and different first- and second-order derivative features to recognize sign language. Wong et al. (2007) uses PCA on motion gradient images of a sequence to obtain features for a Bayesian classifier. To extract motion features, Cooper et al. (2011) extends Haar-like features from spatial domain to spatio-temporal domain and proposes volumetric Haar-like features.

The features introduced above are usually extracted from RGB images captured by a traditional optical camera. Due to the nature of optical sensing, the quality of the captured images is sensitive to lighting conditions and cluttered backgrounds, thus the extracted features from RGB images are not robust. In contrast, depth information from a calibrated camera pair (Rauschert et al., 2002) or direct depth sensors such as LiDAR (Light Detection and Ranging) is more robust to noises and illumination changes. More importantly, depth information is useful for discovering the distance between the hands and body orthogonal to the image plane, which is an important cue for distinguishing some ambiguous gestures. Because the direct depth sensors are expensive, inexpensive depth cameras, e.g., Microsoft's Kinect, have been recently used in gesture recognition (Ershaed et al., 2011; Wu et al., 2012b). Although the skeleton information offered by Kinect is more effective in the expression of human actions than pure depth data, there are some cases that skeleton cannot be extracted correctly, such as interaction between human body and other objects. Actually, in the CHALERAN gesture challenge (Guyon et al., 2013), the skeleton information is not allowed to use. To extract more robust features from Kinect depth images for gesture recognition, Ren et al. (2013) proposes the part based finger shape features, which do not depend on the accurate segmentation of the hands. Wan et al. (2013, 2014b) extend SIFT to spatio-temporal domain and propose 3D EMoSIFT and 3D SMoSIFT to extract features from RGB and depth images, which are invariant to scale and rotation, and have more compact and richer visual representations. Wan et al. (2014a) proposes a discriminative dictionary learning method on 3D EMoSIFT features based on mutual information and then uses sparse reconstruction for classification. Based on 3D Histogram of Flow (3DHOF) and Global Histogram of Oriented Gradient (GHOG), Fanello et al. (2013) applies adaptive sparse coding to capture high-level feature patterns. Wu et al. (2012a) utilizes both RGB and depth information from Kinect and an extended-MHI representation is adopted as the motion descriptors.

The performance of a gesture recognition method is not only related to the used features but also to the adopted classifiers. Many classifiers can be used for gesture recognition, e.g., Dynamic Time Warping (DTW) (Reyes et al., 2011; Lichtenauer et al., 2008; Sabinas et al., 2013), linear SVMs (Fanello et al., 2013), neuro-fuzzy inference system networks (Al-Jarrah and Halawani, 2001), hyper rectangular composite NNs (Su, 2000), and 3D Hopfield NN (Huang and Huang, 1998). Due to the ability of modeling temporal signals, Hidden Markov Model (HMM) is possibly the most well known classifier for gesture recognition. Bauer (Bauer and Kraiss, 2002) proposes a 2D motion model and performs gesture recognition with HMM. Vogler (2003) presents a parallel HMM algorithm to model gestures, which can recognize continuous gestures. Fang et al. (2004) proposes a self-organizing feature maps/hidden Markov model (SOFM/HMM) for gesture recognition in which SOFM is used as an implicit feature extractor for continuous HMM. Recently, Wan et al. (2012a) proposes ScHMM to deal with the gesture recognition where sparse coding is adopted to find succinct representations and Lagrange dual is applied to obtain a codebook.

2.2 One-shot Learning Gesture Recognition and Gesture Characters

Although a large number of work has been done, gesture recognition is still very challenging and has been attracting increasing interests. One motivation is to overcome the well-known overfitting problem when training samples are insufficient. The other one is to further improve gesture recognition by developing novel features and classifiers.

In the case of training samples being insufficient, most of classification methods are very likely to overfit. Therefore, developing gesture recognition methods that use only a small training data set is necessary. An extreme example is the one-shot learning that uses only one training sample per class for training. The proposed work in this paper is also for one-shot learning. In the literature, several previous work has been focused on one-shot learning. In Lui (2012a), gesture sequences are viewed as third-order tensors and decomposed to three Stiefel Manifolds and a natural metric is inherited from the factor manifolds. A geometric framework for least square regression is further presented and applied to gesture recognition. Mahbub et al. (2013) proposes a space-time descriptor and applies Motion History Imaging (MHI) techniques to track the motion flow in consecutive frames. The Euclidean distance based classifiers is used for gesture recognition. Seo and Milanfar (2011) presents a novel action recognition method based on space-time locally adaptive regression kernels and the matrix cosine similarity measure. Malgireddy et al. (2012) presents an end-to-end temporal Bayesian framework for activity classification. A probabilistic dynamic signature is created for each activity class and activity recognition becomes a problem of finding the most likely distribution to generate the test video. Escalante et al. (2013) introduces principal motion components for one-shot learning gesture recognition. 2D maps of motion energy are obtained per each pair of consecutive frames in a video. Motion maps associated to a video are further processed to obtain a PCA model, which is used for gesture recognition with a reconstruction-error approach. More one-shot learning gesture recognition methods are summarized by Guyon et al. (2013).

The intrinsic difference between gesture recognition and other recognition problems is that gesture communication is highly complex and owns its unique characters. Therefore, it is crucial to develop specified features and classifiers for gesture recognition by exploring the unique characters of gestures as explained in Section 1. There are some efforts toward this direction and some work has modeled the component concurrent or sequential organization and achieved significant progress. To capture meaningful linguistic components of gestures, Vogler and Metaxas (1999) proposes PaHMMs which models the movement and shape of user's hands in independent channels and then put them together at the recognition stage. Chen and Koskela (2013) uses multiple Extreme Learning Machines (ELMs) (Huang et al., 2012) as classifiers for simultaneous components. The outputs from the multiple ELMs are then fused and aggregated to provide the final classification results. Chen and Koskela (2013) proposes a novel representation of human gestures and actions based on component concurrent character. They learn the parameters of a statistical distribution that describes the location, shape, and motion flow. Inspired by the sequential organization character of gestures, Wang et al. (2002) uses the segmented subsequences instead of the whole gesture sequence as the basic units that convey the specific semantic expression for the gesture and encode the gesture based on these units. It is successfully applied in large vocabulary sign gestures recognition.

To our best knowledge, there is no work in the literature modeling both the component concurrent character and the sequential organization character in gesture recognition, especially for one-shot learning gesture recognition. It should be noted that these two characters demonstrate the essences of gestures from spatial and temporal aspects, respectively. Therefore, the proposed method that exploits both these characters in a multi-layered framework is desirable to improve gesture recognition.

Test	*Avg. Acc. (%)*	*Identification Strategy*	*Description*
1	75.0	None	Memorizing all the training gestures, and identifying test gesture by recollection
2	90.3	Motion	Drawing lines to record motion direction of each training gesture
3	83.5	Shape	Drawing sketches to describe the hand shape of each training gesture
4	87.6	Location	Drawing sketches to describe the location of each training gesture
5	95.3	Motion & Shape	Strategy 2 and 3
6	100.0	Motion & Location & Shape	Strategy 2, 3 and 4

Table 1: Observations on CGD 2011.

3. Multi-layered Gesture Recognition

The proposed multi-layered classification framework for one-shot learning gesture recognition contains three layers as shown in Figure 1. In the first layer, an improved principle motion is applied to model the motion component. In the second layer, a particle based descriptor is proposed to extract dynamic gesture information and then a weighted dynamic time warping is proposed for the location component classification. In the last layer, we extract unclosed shape contour from the key frame of a gesture sequence. Spatial path warping is further proposed to recognize the shape component. Once the motion component classification at the first layer is accomplished, the original gesture candidates are divided into possible gesture candidates and impossible gesture candidates. The possible gesture candidates are then fed to the second layer which performs the location component classification. Compared with the original gesture candidates, classifying the possible gesture candidates is expected to reduce the computational complexity of the second layer distinctly. The possible gesture candidates are further reduced by the second layer. In the reduced possible gesture candidates, if the first two best matched candidates are difficult to be discriminated, i.e., the absolute difference of their matching scores is lower than a predefined threshold, then the reduced gesture candidates are forwarded to the third layer; otherwise the best matched gesture is output as the final recognition result.

In the remaining of this section, the illuminating cues are first observed in Section 3.1. Inter-gesture segmentation is then introduced in Section 3.2. The motion, location and shape component classifiers in each layer are finally introduced in Section 3.3, Section 3.4 and Section 3.5, respectively.

3.1 Gesture Meaning Expressions and Illuminating Cues

Although from the point of view of gesture linguistics, the basic components and how gestures convey meaning are given (Stokoe, 1960), there is no reference to the importance and complementarity of the components in gesture communication. This section wants to draw some illuminating cues from observations. For this purpose, 10 undergraduate volunteers are invited to take part in the observations.

Five batches of data are randomly selected from the development data of CGD 2011. The pre-defined identification strategies are shown in Table 1. In each test, all the volunteers are asked to follow these identification strategies. For example, in Test 2, they are required to only use the motion cue and draw simple lines to record the motion direction of each gesture in the training set. Then the test gestures are shown to the volunteers to be identified using these drawn lines. The results are briefly summarized in Table 1.

From the observations above, the following illuminating cues can be drawn:

- During gesture recognition, gesture components in the order of importance are motion, location and shape.
- Understanding a gesture requires the observation of all these gesture components. None of these components can convey the complete gesture meanings independently. These gesture components complement each other.

3.2 Inter-gesture Segmentation Based on Movement Quantity

The inter-gesture segmentation is used to segment a multi-gesture sequence into several gesture sequences.[1] To perform the inter-gesture segmentation, we first measure the quantity of movement for each frame in a multi-gesture sequence and then threshold the quantity of movement to get candidate boundaries. Then, a sliding window is adopted to refine the candidate boundaries to produce the final boundaries of the segmented gesture sequences in a multi-gesture sequence.

3.2.1 Quantity of Movement

In a multi-gesture sequence, each frame has the relevant movement with respect to its adjacent frame and the first frame. These movements and their statistical information are useful for inter-gesture segmentation. For a multi-gesture depth sequence I, the Quantity of Movement (QOM) for frame t is defined as a two-dimensional vector

$$QOM(I,t) = [QOM_{Local}(I,t), QOM_{Global}(I,t)] \ ,$$

where $QOM_{Local}(I,t)$ and $QOM_{Global}(I,t)$ measure the relative movement of frame t respective to its adjacent frame and the first frame, respectively. They can be computed as

$$\begin{aligned} QOM_{Local}(I,t) &= \sum_{m,n} \sigma(I_t(m,n), I_{t-1}(m,n)) \ , \\ QOM_{Global}(I,t) &= \sum_{m,n} \sigma(I_t(m,n), I_1(m,n)) \ , \end{aligned}$$

where (m,n) is the pixel location and the indicator function $\sigma(x,y)$ is defined as

$$\sigma(x,y) = \begin{cases} 1 & if|x-y| \geq Threshold_{QOM} \\ 0 & \text{otherwise} \end{cases} \ ,$$

where $Threshold_{QOM}$ is a predefined threshold, which is set to 60 empirically in this paper.

3.2.2 Inter-gesture Segmentation

We assume that there is a home pose between a gesture and another one in a multi-gesture sequence. The inter-gesture segmentation is facilitated by the statistical characteristics of QOM_{Global} of the beginning and ending phases of the gesture sequences in the training

[1] In this paper, we use the term “gesture sequence” to mean an image sequence that contains only one complete gesture and “multi-gesture sequence” to mean an image sequence which may contain one or multiple gesture sequences.

data. One advantage of using QOM_{Global} is that it does not need to segment the user from the background.

Firstly the average frame number L of all gestures in the training set is obtained. The mean and standard deviation of QOM_{Global} of the first and last $\lceil L/8 \rceil$ frames of each gesture sequence are computed. After that, a threshold $Threshold_{inter}$ is obtained as the sum of the mean and the doubled standard deviation. For a test multi-gesture sequence T which has t_s frames, the inter-gesture boundary candidate set is defined as

$$B^{ca}_{inter} = \{i | QOM_{Global}(T, i) \leq Threshold_{inter}, i \in \{1, \cdots, t_s\}\} \ .$$

The boundary candidates are further refined through a sliding window of size $\lceil L/2 \rceil$, defined as $\{j+1, j+2, \cdots, j+\lceil L/2 \rceil\}$ where j starts from 0 to $t_s - \lceil L/2 \rceil$. In each sliding window, only the candidate with the minimal QOM_{Global} is retained and other candidates are eliminated from B^{ca}_{inter}. After the sliding window stops, the inter-gesture boundaries are obtained, which are exemplified as the blue dots in Figure 2. The segmented gesture sequences will be used for motion, location, and shape component analysis and classification.

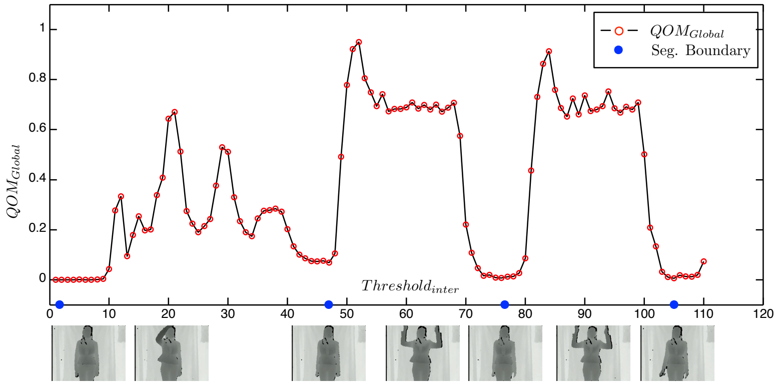

Figure 2: An example of illustrating the inter-gesture segmentation results.

3.3 Motion Component Analysis and Classification

Owing to the relatively high importance of the motion component, it is analyzed and classified in the first layer. The principal motion (Escalante and Guyon, 2012) is improved by using the overlapping block partitioning to reduce the errors of motion pattern mismatchings. Furthermore, our improved principal motion uses both the RGB and depth images. The gesture candidates outputted by the first layer is then fed to the second layer.

3.3.1 Principal Motion

Escalante and Guyon (2012) uses a set of histograms of motion energy information to represent a gesture sequence and implements a reconstruction based gesture recognition method based on principal components analysis (PCA). For a gesture sequence, motion energy images are calculated by subtracting consecutive frames. Thus, the gesture sequence with N frames is associated to $N-1$ motion energy images. Next, a grid of equally spaced blocks is defined over each motion energy image as shown in Figure 3(c). For each motion energy image, the average motion energy in each of the patches of the grid is computed by averaging values of pixels within each patch. Then a 2D motion map for each motion energy image is obtained and each element of the map accounts for the average motion energy of the block centered on the corresponding 2D location. The 2D map is then vectorized into an N_b-dimensional vector. Hence, an N frame gesture sequence is associated to a matrix Y of dimensions $(N-1) \times N_b$. All gestures in the reference set with size V can be represented with matrices Y_v, $v \in \{1, \cdots, V\}$ and PCA is applied to each Y_v. Then the eigenvectors corresponding to the top c eigenvalues form a set $\mathcal{W}_v$, $v = \{1, \cdots, V\}$.

In the recognition stage, each test gesture is processed as like training gestures and represented by a matrix S. Then, S is projected back to each of the V spaces induced by $\mathcal{W}_v$, $v \in \{1, \cdots, V\}$. The V reconstructions of S are denoted by $R_1, \cdots, R_{\mathcal{V}}$. The reconstruction error of each R_v is computed by

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

where n and m are the number of rows and columns of S. Finally, the test gesture is recognized as the gesture with label obtained by $\arg\min_v \varepsilon(v)$.

3.3.2 Improved Principle Motion

Gestures with large movements are usually performed with significant deformation as shown in Figure 3. In Escalante and Guyon (2012), motion information is represented by a histogram whose bins are related to spatial positions. Each bin is analyzed independently and the space interdependency among the neighboring bins is not further considered. The interdependency can be explored to improve the robustness of representing the gesture motion component, especially for the gestures with larger movement. To this end, an overlapping neighborhood partition is proposed. For example, if the size of bins is 20 × 20, the overlapping neighborhood contains 3 × 3 equally spaced neighboring bins in a 60 × 60 square region. The averaged motion energy in the square region is taken as the current bin's value as shown in Figure 3.

The improved principle motion is applied to both the RGB and depth data. The RGB images are transformed into gray images before computing their motion energy images. For each reference gesture, the final V reconstruction errors are obtained by multiplying the reconstruction errors of the depth data and the gray data. These V reconstruction errors are further clustered by K-means to get two centers. The gesture labels associated to those reconstruction errors belonging to the center with smaller value are treated as the possible gesture candidates. The remaining gesture labels are treated as the impossible gesture candidates. Then the possible candidates are fed to the second layer.

We compare the performance of our improved principal motion model with the original principal motion model (Escalante and Guyon, 2012) on the first 20 development batches of CGD 2011. Using the provided code (Guyon et al., 2014; Escalante and Guyon, 2012) as baseline, the average Levenshtein distances (Levenshtein, 1966) are 44.92% and 38.66% for the principal motion and the improved principal motion, respectively.

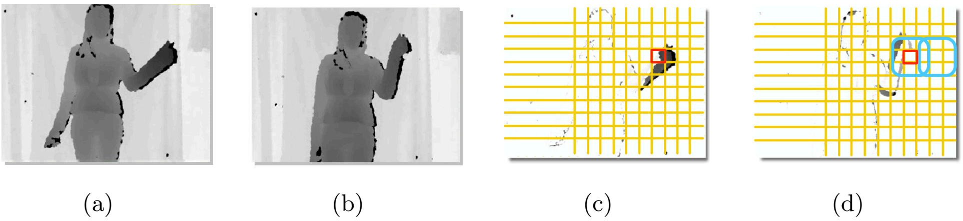

Figure 3: An example of a gesture with large movements. (a) and (b): two frames from a gesture. (c): the motion energy image of (a). The grid of equally spaced bins adopted by the Principle Motion (Escalante and Guyon, 2012). (d): the motion energy image of (b). The overlapped grid used by our method where the overlapping neighborhood includes all 3 × 3 equally spaced neighbor bins.

3.4 Location Component Analysis and Classification

Gesture location component refers to the positions of the arms and hands relative to the body. In the second layer, the sequential organization character of gestures is utilized in the gesture sequence alignment. According to the movement-hold model, each gesture sequence is segmented into semantic units, which convey the specific semantic meanings of the gesture. Accordingly, when aligning a reference gesture and a test gesture, the semantic units are aligned first, then the frames in each semantic unit are registered. A particle-based representation for the gesture location component is proposed to describe the location component of the aligned frames and a Weighted Dynamic Time Warping (WDTW) is proposed for the location component classification.

3.4.1 Intra-gesture Segmentation and Alignment

To measure the distance between location components of a reference gesture sequence $R = \{R_1, R_2 \cdots, R_{L_R}\}$ and a test gesture sequence $T = \{T_1, T_2 \cdots, T_{L_T}\}$, an alignment $\Gamma = \{(i_k, j_k) | k = 1, \cdots, K, i_k \in \{1, \cdots, L_R\}, j_k \in \{1, \cdots, L_T\}\}$ can be determined by the best path in the Dynamic Time Warping (DTW) grid and K is the path length. Then the dissimilarity between two gesture sequences can be obtained as the sum of the distances between the aligned frames.

The above alignment does not consider the sequential organization character of gestures. The movement-hold model proposed by Liddell and Johnson (1989) reveals sequential organization of gestures, which should be explored in the analysis and classification of gesture location component. $QOM_{Local}(I, t)$, described in Section 3.2.1, measures the movement between two consecutive frames. A large $QOM_{Local}(I, t)$ indicates that the t-th frame is in a movement phase, while a small $QOM_{Local}(I, t)$ indicates that the frame is in a hold phase. Among all the frames in a hold phase, the one with the minimal $QOM_{Local}(I, t)$ is the most representative frame and is marked as an anchor frame. Considering the sequential organization character of gestures, the following requirement should be satisfied to compute Γ: each anchor frame in a test sequence must be aligned with one anchor frame in the reference sequence.

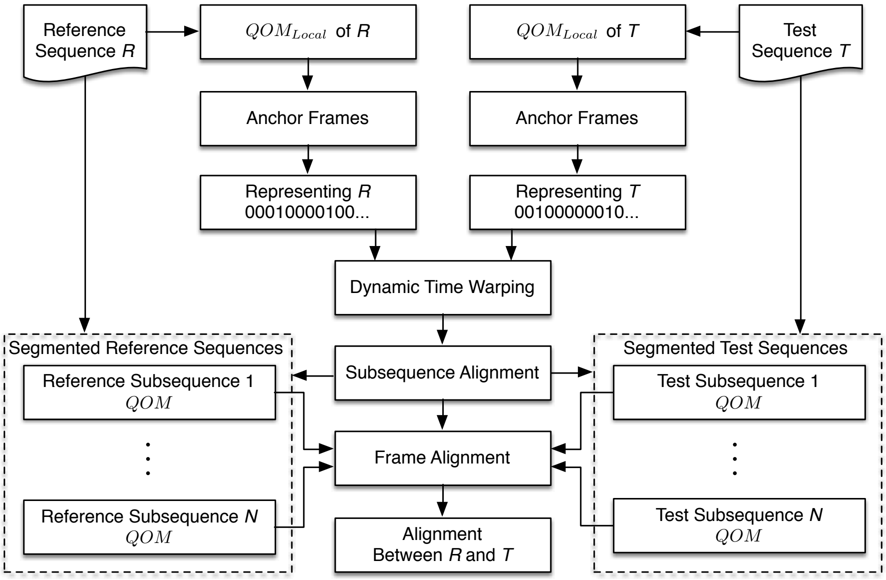

Figure 4: Intra-gesture segmentation and the alignment between test and reference sequences.

As shown in Figure 4, the alignment between the test and reference sequences has two stages. In the first stage, DTW is applied to align the reference and test sequences. Each anchor frame is represented by “1” and the remaining frames are represented by “0”. Then the associated best path $\widehat{\Gamma} = \{(\widehat{i_k}, \widehat{j_k}) | k = 1, \cdots, \widehat{K}\}$ in the DTW grid can be obtained. For each $(\widehat{i_k}, \widehat{j_k})$, if both $\widehat{i_k}$ and $\widehat{j_k}$ are anchor frames, then $\widehat{i_k}$ and $\widehat{j_k}$ are the boundaries of the semantic units. According to the boundaries, the alignment between semantic units of the reference and test sequences is obtained. In the second stage, as shown in Figure 4, each frame in a semantic unit is represented by $[QOM_{Local}, QOM_{Global}]$ and DTW is applied to align the semantic unit pairs separately. Then the final alignment Γ is obtained by concatenating the alignments of the semantic unit pairs.

3.4.2 Location Component Segmentation and its Particle Representation

After the frames of the test and reference sequences are aligned, the next problem is how to represent the location information in a frame. Dynamic regions in each frame contain the most meaningful location information, which are illustrated in Figure 5(i).

A simple thresholding-based foreground-background segmentation method is used to segment the user in a frame. The output of the segmentation is a mask frame that indicates which pixels are occupied by the user as shown in Figure 5(b). The mask frame is then denoised by a median filter to get a denoised frame as shown in Figure 5(c). The denoised frame is first binarized and then dilated with a flat disk-shaped structuring element with radius 10 as shown in Figure 5(d). The swing frame as shown in Figure 5(h) is obtained by subtracting the binarized denoised frame from the dilated frame. The swing region (those

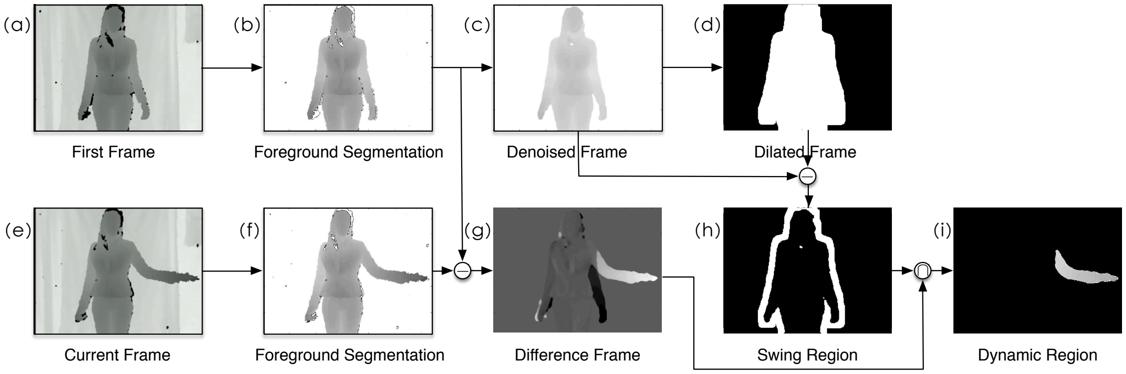

Figure 5: Dynamic region segmentation.

white pixels in the swing frame) covers the slight swing of user's trunk and can be used to eliminate the influence of body swing. From frame t, define set Ξ as

$$\{(m,n)|F_1(m,n) - F_t(m,n) \geq Threshold_{QOM}\} \ ,$$

where F_1 and F_t are the user masks of the first frame and frame t, respectively. $Threshold_{QOM}$ is the same as in Section 3.2.1. For each connected region in Ξ, only if the number of pixels in this region exceeds N_p and the proportion overlapped with swing region is less than r, it is regarded as a dynamic region. Here $N_p = 500$ is a threshold used to remove the meaningless connected regions in the difference frame as shown in Figure 5(g). If a connected region has less than N_p pixels, we think this region should not be a good dynamic region for extracting location features, e.g., the small bright region on the right hand of the user in Figure 5(g). This parameter can be set intuitively. The parameter $r = 50\%$ is also a threshold used to complement with N_p to remove the meaningless connected regions in the difference frame. After using N_p to remove some connected regions, there may be a retained connected region which has more than N_p pixels but it may still not be a meaningful dynamic region for extracting position features if the connected region is caused by the body swing. Obviously we can exploit the swing region to remove such a region. To do this, we first compute the overlap rate between this region and the swing region. If the overlap rate is larger than r, it is reasonable to think this region is mainly produced by the body swing. Therefore, it should be further removed. As like N_p, this parameter is also very intuitive to set and is not very sensitive to the performance.

To represent the dynamic region of frame t, a particle-based description is proposed to reduce the matching complexity. The dynamic region of frame t can be represented by a 3D distribution: $P_t(x,y,z)$ where x and y are coordinates of a pixel and $z = I_t(x,y)$ is the depth value of the pixel. In the form of non-parametric representation, $P_t(x,y,z)$ can be represented by a set of $\widehat{N}$ particles, $P_{Location}(I_t) = \{(x_n, y_n, z_n)|_{n=1}^{\widehat{N}}\}$. We use K-means to cluster all pixels inside the dynamic region into $\widehat{N}$ clusters. Note that for a pixel, both its spatial coordinates and depth value are used. Then the centers of clusters are used as the representative particles. In this paper, 20 representative particles are used for each frame, as shown in Figure 6.

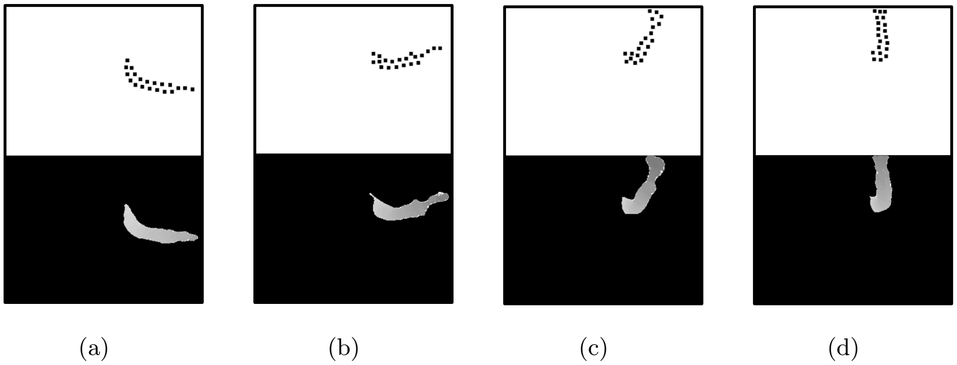

Figure 6: Four examples of particle representation of the location component (the black dots are the particles projected onto X-Y plane).

3.4.3 Location component Classification

Assume the location component of two aligned frames can be represented as two particle sets, $P = \{P_1, P_2 \cdots P_{\widehat{N}}\}$ and $Q = \{Q_1, Q_2 \cdots Q_{\widehat{N}}\}$. The matching cost between particle P_i and Q_j, denoted by $C(P_i, Q_j)$, is computed as their Euclidean distance. The distance of the location component between these two aligned gesture frames is defined by the minimal distance between P and Q. Computing the minimal distance between two particle sets is indeed to find an assignment Π to minimize the cost summation of all particle pairs

$$\Pi = \arg\min_{\Pi} \sum_{i=1}^{\widehat{N}} C(P_i, Q_{\Pi(i)}) \ . \tag{1}$$

This is a special case of the weighted bipartite graph matching and can be solved by the Edmonds method (Edmonds, 1965). Edmonds method which finds an optimal assignment for a given cost matrix is an improved Hungarian method (Kuhn, 1955) with time complexity $O(n^3)$ where n is the number of particles. Finally, the distance of the location component between two aligned gesture frames is obtained

$$dis(P, Q) = \sum_{i=1}^{\widehat{N}} C(P_i, Q_{\Pi(i)}) \ .$$

The distance between the reference sequence R and the test sequence T can be computed as the sum of all distance between the location components of the aligned frames in Γ

$$DIS_{Location}(R, T|\Gamma) = \sum_{k=1}^{K} dis(P_{Location}(R_{i_k}), P_{Location}(T_{j_k})) \ . \tag{2}$$

This measurement implicitly gives all the frames the same weights. However, in many cases gestures are distinguished by only a few frames. Therefore, rather than directly computing

Equation 2, we propose the Weighted DTW (WDTW) to compute the distance of location component between R and T as

$$WDIS_{Location}(R,T|\Gamma) = \sum_{k=1}^{K} W_{i_k}^{R} \times dis(P_{Location}(R_{i_k}), P_{Location}(T_{j_k})) \ ,$$

where $W^R = \{W_{i_k}^R | i_k \in \{1, \cdots, L_R\}\}$ is the weight vector. Different from the method of evaluating the phase difference between the test and reference sequences (Jeong et al., 2011) and the method of assigning different weights to features (Reyes et al., 2011), we assign different weights to the frames of the reference gesture sequence. For each reference gesture sequence, firstly we use the regular DTW to calculate and record the alignment Γ between the current reference gesture sequence and all the other reference gesture sequences. Secondly for each frame in the current reference gesture sequence, we accumulate its corresponding distances with the matched frames in the best path in the DTW. Then, the current frame is weighted by the average distance between itself and all the corresponding frames in the best path. The detailed procedure of computing the weight vector are summarized in Algorithm 1.

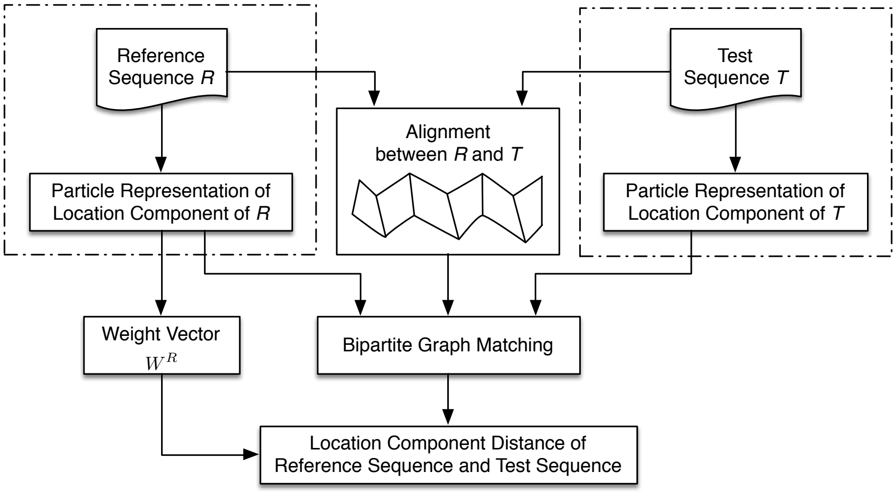
Reference Sequence *R*
Test Sequence *T*
Alignment between *R* and *T*
Particle Representation of Location Component of *R*
Particle Representation of Location Component of *T*
Weight Vector W^R
Bipartite Graph Matching
Location Component Distance of Reference Sequence and Test Sequence

Figure 7: Weighted Dynamic Time Warping framework.

In the second layer, we first use K-means to cluster the input possible gesture candidates into two cluster centers according to the matching scores between the test gesture sequence and the possible gesture candidates. The candidates in the cluster with smaller matching score are discarded. In the remaining candidates, if the first two best matched candidates are difficult to be distinguished, i.e., the absolute difference of their normalized location component distances is lower than a predefined threshold ϵ, then these candidates are forwarded to the third layer; otherwise the best matched candidate is output as the final recognition result. Two factors influence the choice of the parameter ϵ. The first one is the number of the gesture candidates and the other one is the type of gestures. When the number of the gesture candidates is large or most of the gesture candidates are the shape

Algorithm 1 Computing weight vector W^R for a reference R

```
Input: all the $O$ reference gesture depth sequences: $I^1, I^2, \cdots, I^O$
Output: weight vector for $R$, $W^R = \{W^R_m | m \in \{1, \cdots, L_R\}\}$
for each $m \in [1, L_R]$ do
    $W^R_m = 0$
    $N^R_m = 0$
end for
for each $n \in [1, O]$ do
    Compute the alignment $\Gamma = \{(i_k, j_k)\}$ between $R$ and $I^n$
    for each $m \in [1, L_R]$ do
        $W^R_m = W^R_m + \sum_{(i_k=m, j_k)\in\Gamma} dis(P_{Location}(R_{i_k}), P_{Location}(I^n_{j_k}))$
        $N^R_m = N^R_m + \sum_{(i_k, j_k)\in\Gamma} \delta(i_k = m)$
        if $n = O$ then
            $W^R_m = W^R_m \big/ N^R_m$
        end if
    end for
end for
```

dominant gestures, a high threshold is preferred. In our experiments, we empirically set its value with 0.05 by observing the matching scores between the test sample and each gesture candidates.

3.5 Shape Component Analysis and Classification

The shape in a hold phase is more discriminative than the one in a movement phase. The key frame in a gesture sequence is defined as the frame which has the minimization QOM_{Local}. Shape component classifier classifies the shape features extracted from the key frame of a gesture sequence using the proposed Spatial Path Warping (SPW), which first extracts unclosed shape context (USC) features and then calculates the distance between the USCs of the key frames in the reference and the test gesture sequences. The test gesture sequence is classified as the gesture whose reference sequence has the smallest distance with the test gesture sequence.

3.5.1 Unclosed Shape Segmentation

The dynamic regions of a frame have been obtained in Section 3.4.2. In a key frame, the largest dynamic region D is used for shape segmentation. Although shapes are complex and do not have robust texture and structured appearance, in most cases shapes can be distinguished by their contours. The contour points of D are extracted by the Canny algorithm. The obtained contour point set is denoted by C_1 as shown in Figure 8(a). K-means is adopted to cluster the points in D into two clusters based on the image coordinates and depth of each point. If a user faces to the camera, the cluster with smaller average depth contains most of information for identifying the shape component. Canny algorithm is used again to extract contour points of the cluster with smaller average depth. The obtained closed contour point set is denoted by C_2 as shown in Figure 8(b). Furthermore, an unclosed contour point set can be obtained by $C_3 = C_2 \bigcap C_1$ as shown in Figure 8(c), which will be used to reduce the computational complexity of matching shapes.

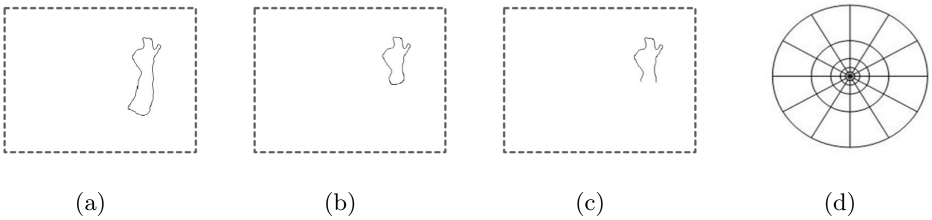

Figure 8: Unclosed shape segmentation and context representation. (a) is an example of point set C_1, (b) is an example of point set C_2 and (c) is an example of obtained point set C_3; (d) is the log-polar space used to decide the ranges of K bins.

3.5.2 Shape Representation and Classification

The contour of a shape consists of a 2-D point set $\mathbb{P} = \{p_1, p_2, \cdots, p_N\}$. Their relative positions are important for the shape recognition. From the statistical point of view, Belongie et al. (2002) develops a strong shape contour descriptor, namely Shape Context (SC). For each point p_i in the contour, a histogram h_{p_i} is obtained as the shape context of the point whose k-th bin is calculated by

$$h_{p_i}(k) = \sharp\{(p_j - p_i) \in bin(k) | p_j \in \mathbb{P}, i \neq j, k \in \{1, \cdots, K\}\} ,$$

where $bin(k)$ defines the quantification range of the k-th bin. The log-polar space for bins is illustrated in Figure 8(d).

Assume $\mathbb{P}$ and $\mathbb{Q}$ are the point sets for the shape contours of two key frames, the matching cost $\Phi(p_i, q_j)$ between two points $p_i \in \mathbb{P}$ and $q_j \in \mathbb{Q}$ is defined as

$$\Phi(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_{p_i}(k) - h_{q_j}(k)]^2}{h_{p_i}(k) + h_{q_j}(k)} .$$

Given the set of matching costs between all pairs of points $p_i \in \mathbb{P}$ and $q_j \in \mathbb{Q}$, computing the minimal distance between $\mathbb{P}$ and $\mathbb{Q}$ is to find a permutation Ψ to minimize the following sum

$$\Psi = \arg\min_{\Psi} \sum_{i} \Phi(p_i, q_{\Psi(i)}) ,$$

which can also be solved by the Edmonds algorithm as like solving Equation 1.

An unclosed contour contains valuable spatial information. Thus, a Spatial Path Warping algorithm (SPW) is proposed to compute the minimal distance between two unclosed contours. Compared with the Edmonds algorithm, the time complexity of the proposed SPW is reduced from $O(n^3)$ to $O(n^2)$ where n is the size of the point set of an unclosed shape contour. As shown in Figure 8(c), the points on an unclosed contour can be represented as a clockwise contour point sequence. SPW is used to obtain the optimal match between two given unclosed contour point sequences. For two unclosed contour point sequences $\{p'_1, \cdots, p'_n\}$, $\{q'_1, \cdots, q'_m\}$, a dynamic window is set to constrain the points that one point can match, which makes the matching more robust to local shape variation. We set the window size w with $\max(L_s, abs(n-m))$. In most cases, the window size is the absolute difference between the lengths of the two point sequences. In extreme cases, if two sequences have very close lengths, i.e., their absolute difference is less then L_s, we set the the window size with L_s. The details of proposed SPW are summarized in Algorithm 2.

Algorithm 2 Computing distance between two unclosed contour point sequences

```
Input: two unclosed contour point sequences $\{p'_1, \cdots, p'_n\}$, $\{q'_1, \cdots, q'_m\}$
Output: distance between these two point sequences $SPW[n, m]$.
Set $w = \max(L_s, abs(n-m))$
for each $i \in [0, n]$ do
    for each $j \in [0, m]$ do
        $SPW[i, j] = \infty$
    end for
end for
$SPW[0, 0] = 0$
for each $i \in [1, n]$ do
    for each $j \in [\max(1, i-w), \min(m, i+w)]$ do
        $SPW[i, j] = \Phi(p'_i, q'_j) + \min(SPW[i-1, j], SPW[i, j-1], SPW[i-1, j-1])$
    end for
end for
```

4. Experiments

In this section, extensive experiment results are presented to evaluate the proposed multi-layered gesture recognition method. All the experiments are performed in Matlab 7.12.0 on a Dell PC with Duo CPU E8400. The ChaLearn Gesture Dataset (CGD 2011) (Guyon et al., 2014) is used in all experiments, which is designed for one-shot learning. The CGD 2011 consists of 50,000 gestures (grouped in 500 batches, each batch including 47 sequences and each sequence containing 1 to 5 gestures drawn from one of 30 small gesture vocabularies of 8 to 15 gestures), with frame size 240 × 320, 10 frames/second, recorded by 20 different users.

The parameters used in the proposed method are listed in Table 2. Noted that the parameters c and N_b are set with the default values used in the sample code of the principal model.[2] The threshold for foreground and background segmentation is adaptively set to the maximal depth minus 100 for each batch data. For example, the maximal depth of the devel01 batch is 1964. Then the threshold for this batch is 1864. The number 100 is in fact a small bias from the maximal depth, which is empirically set in our experiments. We observed that slightly changing this number does not significantly affect the segmentation. Considering the tradeoff between the time complexity and recognition accuracy, in our experiments, we empirically set $\widehat{N}$ to 20, which achieves the desired recognition performance.

In our experiments, Levenshtein distance is used to evaluate the gesture recognition performance, which is also used in the CHALERAN gesture challenge. It is the minimum number of edit operations (substitution, insertion, or deletion) that have to be performed from one sequence to another (or vice versa). It is also known as "edit distance".

[2] The code is available at `http://gesture.chalearn.org/data/sample-code`

Parameter and Description	*Applied to*	*Value*	*From Prior or Not*	*Sensitive to Performance*	*Training Data Used or Not*
N_p: Minimal number of pixels in a connected region	**D**	500	Y	N	Y
r: Maximal overlap rate between a connected region and the swing region	D	50%	N	N	N
ϵ: Threshold for the difference between the first two largest matches	**D**, **E**	0.05	Y	N	Y
L_s: Minimal length of the sliding window	**E**	5	N	N	N
$Threshold_{QOM}$	**A**, **D**, **E**	60	Y	Y	N
$Threshold_{inter}$	**A**	adaptive	N	Y	Y
c: number of eigenvalues for each gesture	**C**	10	Y	N	N
N_b: number of bins for each motion energy image	**C**	192	Y	N	N
$\widehat{N}$: number of particles	**D**	20	Y	N	N
Threshold for foreground and background segmentation	**D**, **E**	Max depth - 100	Y	N	Y

A: Inter-gesture segmentation; **B**: intra-gesture segmentation; **C**: Motion component analysis and classification
D: Location component analysis and classification; **E**: Shape component analysis and classification; **Training data**: CGD 2011

Table 2: The parameters used in the proposed multi-layered gesture recognition and their descriptions.

4.1 Performance of Our Method with Different Layers

We evaluate the performance of the proposed method with different layers on the development (devel01 ∼ devel480) batches of CGD 2011 and Table 3 reports the results. If only the first layer is used for classification, the average Levenhstein distance is 37.53% with running time 0.54 seconds per gesture. If only the second layer is used for recognition, the average Levenhstein distance is 29.32% with running time 6.03 seconds per gesture. If only the third layer is used, the average Levenhstein distance is 39.12% with the running time 6.64 seconds per gesture. If the first two layers are used, the average Levenhstein distance is 24.36% with running time 2.79 seconds per gesture. If all three layers are used, the average normalized Levenhstein distance is 19.45% with running time 3.75 seconds per gesture.

methods	***First layer for recognition***	***Second layer for recognition***	***Third layer for recognition***	***First two layers for recognition***	***Three layers for recognition***
TeLev (%)	37.53	29.32	39.12	24.36	19.45
Recognition time per gesture (s)	0.54	6.03	6.64	2.79	3.75

Table 3: Performance of using the first layer, the second layer, the third layer, first two layers and three layers on ChaLearn gesture data set (devel01 ∼ devel480).

From these comparison results, we can see that the proposed method achieves high recognition accuracy while having low computational complexity. The first layer can identify the gesture candidates at the speed of 80 fps (frames per second). The second layer has relatively high computational complexity. If we only use the second layer for classification, the average computing time is roughly 11 times of the first layer. Despite with relatively high computational cost, the second layer has stronger classification ability. Compared with using only the second layer, the computational complexity of using the first two layers in the proposed method is distinctly reduced and can achieve 16 fps. The reason is that although the second layer is relatively complex, the gesture candidates forwarded to it are significantly reduced by the first layer. When all three layers are used, the proposed method still achieve about 12 fps, which is faster than the video recording speed (10 fps) of CGD 2011.

4.2 Comparison with Recent Representative Methods

We compare the proposed method with other recent representative methods on the first 20 development data batches. Table 4 reports the performance of the proposed method on each batch and also the average performance on all 20 batches. The average performance of the proposed method and the compared methods are shown in Table 5.

Batch	Second layer for recognition		First two layers for recognition		Three layers for recognition	
	TeLev (%)	*Recognize time per gesture (s)*	*TeLev (%)*	*Recognize time per gesture (s)*	*TeLev (%)*	*Recognize time per gesture (s)*
1	7.24	6.78	0.11	3.40	1.11	3.59
2	41.21	11.38	44.21	7.10	34.35	10.00
3	62.98	8.86	69.20	2.99	39.95	5.61
4	4.51	5.98	3.93	2.10	6.93	2.30
5	11.68	10.96	2.62	3.05	4.77	3.31
6	44.64	5.59	39.94	2.69	23.51	3.42
7	12.44	3.59	8.51	1.70	8.51	1.79
8	5.56	4.94	0.00	2.14	5.71	2.94
9	10.56	5.10	6.44	2.50	6.44	3.01
10	44.21	5.88	29.13	3.24	16.52	3.95
11	42.75	6.46	36.36	3.98	28.93	6.31
12	8.56	5.16	1.06	2.00	7.06	2.34
13	16.24	3.68	12.93	1.20	12.93	1.99
14	44.69	2.50	40.13	0.90	27.98	2.35
15	15.78	4.61	4.21	1.09	6.21	2.19
16	36.54	8.35	36.27	4.21	23.41	6.94
17	36.25	9.10	29.55	5.10	26.32	5.39
18	62.4	1.99	69.21	0.81	53.55	1.60
19	54.31	5.07	51.32	2.84	47.61	3.02
20	17.74	2.58	10.61	1.40	10.61	2.01
Average	**29.02**	**5.93**	**24.79**	**2.73**	**19.62**	**3.69**

Table 4: Recognition performance of using the second layer, first two layers and three layers on first 20 development batches of CGD 2011 (TeLev is the average Levenshtein distance).

Methods	*Extend-MHI* *Wu et al. (2012a)*	*Manifold LSR* *Lui (2012a)*	*Sparse Coding* *Fanello et al. (2013)*	*Temporal Bayesian* *Malgireddy et al. (2012)*	*Motion History* *Mahbub et al. (2013)*	*CSMMI+3D EMoSIFT* *Wan et al. (2014a)*	*Proposed*
TeLev (%)	26.00	28.73	25.11	24.09	31.25	18.76	19.62
TeLen	#	6.24	5.02	#	18.01	#	5.91

Table 5: Performance comparison on the 20 development data batches (TeLen is the average error made on the number of gestures).

For the comparison on each batch, the proposed method is compared with a manifold and nonlinear regression based method (Manifold LSR) (Lui, 2012b), an extended motion-history-image and correlation coefficient based method (Extended-MHI) (Wu et al., 2012a), and a motion silhouettes based method (Motion History) (Mahbub et al., 2013). The comparison results are shown in Figure 9.

In batches 13, 14, 17, 18, 19, the proposed method does not achieve the best performance. However, the proposed method achieves the best performance in the remaining 15 batches.

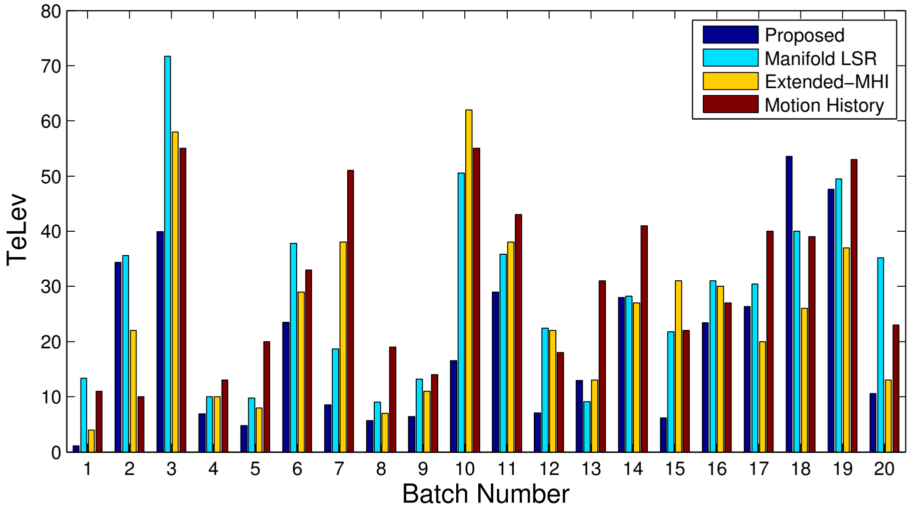

Figure 9: Performance comparison on the 20 development batches in CGD 2011.

In batches 3, 10 and 11, most of gestures consist of static shapes, which can be efficiently identified by the shape classifier in the third layer. Batches 1, 4, 7 and 8 consist of motion dominated gestures, which can be classified by the motion and location component classifiers in the first and second layers. In batches 18 and 19, the proposed method has relatively poor performance. As in batch 18, most of gestures have small motion, similar locations, and non-stationary hand shapes. These gestures may be difficult to be identified by the proposed method. In batch 19, the gestures have similar locations and hands coalescence, which is difficult to be identified by the second layer and the third layer classifiers in our method. Overall, the proposed method significantly outperforms other recent competitive methods.

The proposed method is further compared with DTW, continuous HMM (CHMM), semi-continuous HMM (SCHMM) and SOFM/HMM (Fang et al., 2004) on the development (devel01 $\sim$ devel480) batches of CGD 2011. All compared methods use one of three feature descriptors including dynamic region grid representation (DP), dynamic region particle representation (DG) and Dynamic Aligned Shape Descriptor (DS) (Fornés et al., 2010).

- **Dynamic region grid representation**. For the dynamic region of the current frame obtained in Section 3.4.2, a grid of equally spaced cells is defined and the default size of grid is 12×16. For each cell, the average value of depth in the square region is taken as the value of current bin. So a 12×16 matrix is generated, which is vectorized into the feature vector of the current frame.
- **Dynamic region particle representation**. The particles for the current frame obtained in Section 3.4.2 cannot directly be used as an input feature vector and they have to be reorganized. The 20 particles $\{(x_n, y_n, z_n)|_{n=1}^{20}\}$ are sorted according to $\|(x_n, y_n)\|^2$ and then the sorted particles are concatenated in order to get a 60-dimensional feature vector to represent the current frame.

- **Dynamic region D-Shape descriptor** (Fornés et al., 2010). Firstly, the location of some concentric circles is defined, and for each one, the locations of the equidistant voting points are computed. Secondly, these voting points will receive votes from the pixels of the shape of the dynamic region, depending on their distance to each voting point. By locating isotropic equidistant points, the inner and external part of the shape could be described using the same number of voting points. In our experiment, we used 11 circles for the D-Shape descriptor. Once we have the voting points, the descriptor vector is computed.

Here, each type of HMM is a 3-state left-to-right model allowing possible skips. For CHMM and SCHMM, the covariance matrix is a diagonal matrix with all diagonal elements being 0.2. The comparison results are reported in Table 6.

Method	***Number of Mixtures for each state***	***TeLev (%)***			***Recognition time per gesture (s)***		
		DP	***DG***	***DS***	***DP***	***DG***	***DS***
DTW	#	38.23	41.19	33.16	2.67	2.51	2.60
CHMM	5	31.41	33.29	31.13	6.91	6.83	6.89
SCHMM	30	31.01	32.92	29.35	6.82	6.75	6.79
SOFM/HMM	5	28.27	30.31	27.20	6.77	6.71	6.74

DP: dynamic region particle representation; **DG**: dynamic region grid representation
DS: dynamic region D-Shape descriptor

Table 6: Performance of different sequence matching methods on 480 development batches of CGD 2011.

Compared with these methods, the proposed method achieves the best performance. Noted that in all compared methods, SOFM/HMM classifier with the DS descriptor achieves the second best performance. As explained in Section 1, sequentially modeling motion, position and shape components is very important for improving the performance of gesture recognition. Except the proposed method, other compared methods do not utilize these components. On the other hand, statistical models like CHMM, SCHMM and SOFM/HMM need more training samples to estimate model parameters, which also affect their performance in the one-shot learning gesture recognition.

5. Conclusion

The challenges of gesture recognition lie in the understanding of the unique characters and cues of gestures. This paper proposed a novel multi-layered gesture recognition with Kinect, which is linguistically and perceptually inspired by the phonological model and the movement-hold model. Together with the illuminating cues drawn from observations, the component concurrent character and the sequential organization character of gestures are all utilized in the proposed method. In the first layer, an improved principle motion is applied to model the gesture motion component. In the second layer, a particle based descriptor is proposed to extract dynamic gesture information and then a weighted dynamic time warping is proposed to classify the location component. In the last layer, the spatial path warping is further proposed to classify the shape component represented by unclosed shape context, which is improved from the original shape context but needs lower matching time. The proposed method can obtain relatively high performance for one-shot learning gesture recognition. Our work indicates that the performance of gesture recognition can be significantly improved by exploring and utilizing the unique characters of gestures, which will inspire other researcher in this field to develop learning methods for gesture recognition along this direction.

Acknowledgments

We would like to acknowledge the editors and reviewers, whose valuable comments greatly improved the manuscript. Specially, we would also like to thank Escalante and Guyon who kindly provided us the principal motion source code and Microsoft Asian who kindly provided two sets of Kinect devices. 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 No. 61272386, 61100096 and 61300111.

References

Tushar Agrawal and Subhasis Chaudhuri. Gesture recognition using motion histogram. In *Proceedings of the Indian National Conference of Communications*, pages 438–442, 2003.

Omar Al-Jarrah and Alaa Halawani. Recognition of gestures in Arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1):117–138, 2001.

George Awad, Junwei Han, and Alistair Sutherland. A unified system for segmentation and tracking of face and hands in sign language recognition. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 1, pages 239–242, 2006.

Malek Baklouti, Eric Monacelli, Vincent Guitteny, and Serge Couvet. Intelligent assistive exoskeleton with vision based interface. In *Proceedings of the 5th International Conference On Smart Homes and Health Telematics*, pages 123–135, 2008.

Britta Bauer and Karl-Friedrich Kraiss. Video-based sign recognition using self-organizing subunits. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 434–437, 2002.

Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

Qin Cai, David Gallup, Cha Zhang, and Zhengyou Zhang. 3D deformable face tracking with a commodity depth camera. In *Proceedings of the 11th European Conference on Computer Vision*, pages 229–242, 2010.

Xi Chen and Markus Koskela. Online RGB-D gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, pages 467–474, 2013.

Helen Cooper, Brian Holt, and Richard Bowden. Sign language recognition. In *Visual Analysis of Humans*, pages 539–562, 2011.

Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205–2231, 2012.

Andrea Corradini. Real-time gesture recognition by means of hybrid recognizers. In *Proceedings of International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, pages 34–47, 2002.

Ross Cutler and Matthew Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 416–416, 1998.

Nasser Dardas. *Real-time Hand Gesture Detection and Recognition for Human Computer Interaction*. PhD thesis, University of Ottawa, 2012.

Paul Doliotis, Alexandra Stefan, Chris Mcmurrough, David Eckhard, and Vassilis Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, page 20, 2011.

Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

H Ershaed, I Al-Alali, N Khasawneh, and M Fraiwan. An Arabic sign language computer interface using the Xbox Kinect. In *Proceedings of the Annual Undergraduate Research Conference on Applied Computing*, volume 1, 2011.

Hugo Escalante and Isabelle Guyon. Principal motion. `http://www.causality.inf.ethz.ch/Gesture/principal_motion.pdf`, 2012.

Hugo Jair Escalante, Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Jun Wan. Principal motion components for gesture recognition using a single-example. *arXiv preprint arXiv:1310.4822*, 2013.

Sean Ryan Fanello, Ilaria Gori, Giorgio Metta, and Francesca Odone. One-shot learning for real-time action recognition. In *Pattern Recognition and Image Analysis*, pages 31–40, 2013.

Gaolin Fang, Wen Gao, and Debin Zhao. Large vocabulary sign language recognition based on fuzzy decision trees. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(3):305–314, 2004.

Alicia Fornés, Sergio Escalera, Josep Lladós, and Ernest Valveny. Symbol classification using dynamic aligned shape descriptor. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 1957–1960, 2010.

Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Hugo Jair Escalante, and Ben Hamner. Results and analysis of the Chalearn gesture challenge 2012. In *Proceedings of International Workshop on Advances in Depth Image Analysis and Applications*, pages 186–204, 2013.

Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Hugo Jair Escalante. The chalearn gesture dataset (CGD 2011). *Machine Vision and Applications*, 2014. DOI: 10.1007/s00138-014-0596-3.

Chung-Lin Huang and Wen-Yi Huang. Sign language recognition using model-based tracking and a 3D Hopfield neural network. *Machine Vision and Applications*, 10(5-6):292–307, 1998.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):513–529, 2012.

Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

Timor Kadir, Richard Bowden, Eng Jon Ong, and Andrew Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 1–10, 2004.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966.

Jeroen F Lichtenauer, Emile A Hendriks, and Marcel JT Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, 2008.

Scott K Liddell and Robert E Johnson. American sign language: The phonological base. *Sign Language Studies*, 64:195–278, 1989.

Yui Man Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13(1):3297–3321, 2012a.

Yui Man Lui. A least squares regression framework on manifolds and its application to gesture recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 13–18, 2012b.

Upal Mahbub, Tonmoy Roy, Md Shafiur Rahman, and Hafiz Imtiaz. One-shot-learning gesture recognition using motion history based gesture silhouettes. In *Proceedings of the International Conference on Industrial Application Engineering*, pages 186–193, 2013.

Manavender R Malgireddy, Ifeoma Inwogu, and Venu Govindaraju. A temporal Bayesian model for classifying, detecting and localizing activities in video sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 43–48, 2012.

Manar Maraqa and Raed Abu-Zaiter. Recognition of arabic sign language (ArSL) using recurrent neural networks. In *Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies*, pages 478–481, 2008.

Tin Hninn Hninn Maung. Real-time hand tracking and gesture recognition system using neural networks. *World Academy of Science, Engineering and Technology*, 50:466–470, 2009.

Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

Kai Nickel and Rainer Stiefelhagen. Visual recognition of pointing gestures for human–robot interaction. *Image and Vision Computing*, 25(12):1875–1884, 2007.

Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2011.

Eng-Jon Ong and Richard Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 889–894, 2004.

Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, and Subhashis Banerjee. Recognition of dynamic hand gestures. *Pattern Recognition*, 36(9):2069–2081, 2003.

Ingmar Rauschert, Pyush Agrawal, Rajeev Sharma, Sven Fuhrmann, Isaac Brewer, and Alan MacEachren. Designing a human-centered, multimodal GIS interface to support emergency management. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, pages 119–124, 2002.

Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using Kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, 2013.

Miguel Reyes, Gabriel Dominguez, and Sergio Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1182–1188, 2011.

Yared Sabinas, Eduardo F Morales, and Hugo Jair Escalante. A One-Shot DTW-based method for early gesture recognition. In *Proceedings of 18th Iberoamerican Congress on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 439–446, 2013.

Hae Jong Seo and Peyman Milanfar. Action recognition from one example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):867–882, 2011.

Ling Shao and Ling Ji. Motion histogram analysis based key frame extraction for human action/activity representation. In *Proceedings of Canadian Conference on Computer and Robot Vision*, pages 88–92, 2009.

Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

E Stergiopoulou and N Papamarkos. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8):1141–1158, 2009.

William C Stokoe. Sign language structure: An outline of the visual communication systems of the american deaf. *Studies in Linguistics, Occasional Papers*, 8, 1960.

Mu-Chun Su. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 30(2):276–281, 2000.

Christian Vogler and Dimitris Metaxas. Parallel hidden markov models for american sign language recognition. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 116–122, 1999.

Christian Philipp Vogler. *American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-based Modeling and Parallel Hidden Markov Models.* PhD thesis, University of Pennsylvania, 2003.

J. Wachs, M. Kolsch, H. Stem, and Y. Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

Jun Wan, Qiuqi Ruan, Gaoyun An, and Wei Li. Gesture recognition based on hidden markov model from sparse representative observations. In *Proceedings of the IEEE 11th International Conference on Signal Processing*, volume 2, pages 1180–1183, 2012a.

Jun Wan, Qiuqi Ruan, Gaoyun An, and Wei Li. Hand tracking and segmentation via graph cuts and dynamic model in sign language videos. In *Proceedings of IEEE 11th International Conference on Signal Processing*, volume 2, pages 1135–1138. IEEE, 2012b.

Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14(1): 2549–2582, 2013.

Jun Wan, Vassilis Athitsos, Pat Jangyodsuk, Hugo Jair Escalante, Qiuqi Ruan, and Isabelle Guyon. CSMMI: Class-specific maximization of mutual information for action and gesture recognition. *IEEE Transactions on Image Processing*, 23(7):3152–3165, 2014a.

Jun Wan, Qiuqi Ruan, Wei Li, Gaoyun An, and Ruizhen Zhao. 3D SMoSIFT: Three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos. *Journal of Electronic Imaging*, 23(2):023017, 2014b.

Chunli Wang, Wen Gao, and Shiguang Shan. An approach based on phonemes to large vocabulary Chinese sign language recognition. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, pages 411–416, 2002.

J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

Shu-Fai Wong, Tae-Kyun Kim, and Roberto Cipolla. Learning motion categories using both semantic and structural information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

Di Wu, Fan Zhu, and Ling Shao. One shot learning gesture recognition from RGBD images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12, 2012a.

Shen Wu, Feng Jiang, Debin Zhao, Shaohui Liu, and Wen Gao. Viewpoint-independent hand gesture recognition system. In *Proceedings of the IEEE Conference on Visual Communications and Image Processing*, pages 43–48, 2012b.

Morteza Zahedi, Daniel Keysers, and Hermann Ney. Appearance-based recognition of words in american sign language. In *Proceedings of Second Iberian Conference on Pattern Recognition and Image Analysis*, pages 511–519, 2005.

Liang-Guo Zhang, Yiqiang Chen, Gaolin Fang, Xilin Chen, and Wen Gao. A vision-based sign language recognition system using tied-mixture density HMM. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, pages 198–204, 2004.