Pattern Recognition 44 (2011) 1614–1628

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Model-based segmentation and recognition of dynamic gestures in continuous video streams

Hong Li [a], Michael Greenspan [a,b,*]

[a] Electrical and Computer Engineering, 19 Union Street, Walter Light Hall, Queen's University, Kingston, Ontario, Canada K7L 3N6
[b] School of Computing, 557 Goodwin Hall, Queen's University, Kingston, Ontario, Canada K7L 3N6

ARTICLE INFO

Article history:
Received 23 December 2009
Received in revised form
24 May 2010
Accepted 21 December 2010
Available online 7 January 2011

Keywords:
Continuous gesture recognition
Gesture segmentation
Motion Signature
Gesture Model
Dynamic Programming
Dynamic Time Warping

ABSTRACT

Segmentation and recognition of continuous gestures are challenging due to spatio-temporal variations and endpoint localization issues. A novel multi-scale Gesture Model is presented here as a set of 3D spatio-temporal surfaces of a time-varying contour. Three approaches, which differ mainly in endpoint localization, are proposed: the first uses a motion detection strategy and multi-scale search to find the endpoints; the second uses Dynamic Time Warping to roughly locate the endpoints before a fine search is carried out; the last approach is based on Dynamic Programming. Experimental results on two arm and single hand gestures show that all three methods achieve high recognition rates, ranging from 88% to 96% for the two arm test, with the last method performing best.

1. Introduction

Human gestures are a natural means of interaction and communication among people. Gestures employ hand, limb and body motion to express ideas or exchange information non-verbally. With the efforts to improve the way that humans interact with computers, there has been an increasing interest in trying to integrate human gestures into human–computer interface (HCI) [1–3]. Compared to traditional methods like key-board, mouse or control stick, gestures are an attractive alternative that is more natural and intuitive to humans. Gesture recognition is also important in automated surveillance and human monitoring applications, where they can yield valuable clues into human activities and intentions.

Generally, gestures are captured and embedded in continuous video streams, and a gesture recognition system must have the capability to extract useful information and identify distinct motions automatically. However, there are two issues that are known to be highly challenging for gesture segmentation and recognition: spatio-temporal variation, and endpoint localization [4,5]. Spatio-temporal variation comes from the fact that not only do different people move in different ways, but also even repeated motions by the same subject may vary. Among all the factors contributing to this variation, motion speed is the most influential, which makes the gesture signal demonstrate multiple temporal scales. An ideal gesture recognizer should be robust enough to handle variations over a wide range, especially in the temporal domain. The endpoint localization issue is to determine the start and end time of a gesture in a continuous stream. Just as there are no breaks for each word spoken in speech signals, in most naturally occurring scenarios, gestures are linked together continuously without any obvious pause between individual gestures. Therefore, it is infeasible to determine the endpoints of individual gestures by looking for distinct pauses between gestures. Exhaustively searching through all the possible points is also obviously prohibitively expensive. Many existing methods assume that input data have been segmented into motion units either at the time of capture or manually after capture. This is often referred to as isolated gesture recognition (IGR) and cannot be extended easily to real-world applications requiring the recognition of continuous gestures [6–12].

Although it is possible to regard segmentation and recognition as two separate issues and perform segmentation before recognition by looking into abrupt motion feature changes, some of these features are only present in specific contexts and cannot be generalized to other applications, such as stops between gestures [13], certain movement patterns [14], or abnormal velocity and severe curvature [15]. Indeed, we believe that segmentation and recognition are actually two aspects of the same problem and should be conducted

* Corresponding author at: Electrical and Computer Engineering, 19 Union Street, Walter Light Hall, Queen's University, Kingston, Ontario, Canada K7L 3N6. Tel.: +1 613 533 6000x74949/77736; fax: +1 613 533 6615.
E-mail address: michael.greenspan@queensu.ca (M. Greenspan).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.12.014

simultaneously [16,17,5,18,4,19]. The process of localizing meaningful segments in a continuous stream can be greatly benefitted if a recognition process is available to validate which segments are real gestures among all possible candidates. Once a segment is confirmed to be a predefined gesture in the database, the start and end times of that gesture can either be determined at the same time, or can be obtained easily with subsequent processing.

In this paper, we propose and compare three methods that simultaneously segment and recognize gestures from continuous streams [20–22]. All three methods are based on the same model, i.e., a *Gesture Model*, which captures reliable information of motion and is capable of accommodating a wide range of spatio-temporal variations. To build a Gesture Model, only the contours of the subject are extracted over time, and no detailed human geometrical information is required, which has been proven to be a difficult task in and of itself [7]. Furthermore, as opposed to Hidden Markov Models (HMM), which have been widely used in speech recognition [23] and recently gesture recognition [13,24,16,5,17], our model only requires a relatively small amount data for training, compared to the large training set requirements of HMMs.

Although all three methods perform segmentation and recognition simultaneously, the key difference lies in how the endpoints of gestures are located. While the first method combines a motion detection technique and an explicit multi-scale search to find the start and end times of a gesture, the other two methods further improve the efficiency and accuracy of Method I by employing a Dynamic Programming (DP) [25] technique. In particular, the second algorithm extends Dynamic Time Warping (DTW) [26], which is a special application of DP and can only be applied to isolated gesture recognition due to its endpoint constraint [27,28,12], to the case of continuous gesture recognition. Note that Method II is very different from the traditional Level-Building DTW, which was initially proposed by Myers and Rabiner [29] for connected word recognition, and was recently enhanced for continuous sign recognition [30,31]. In Level-Building DTW, the endpoint constraint of DTW still holds and the endpoints of each gesture can only be obtained after all the gestures in a sequence have been performed, while the proposed Method II is able to recognize gestures immediately after execution. The third method uses a totally different framework based on DP. Instead of estimating the possible endpoints of a gesture and then conducting a fine search as in Method II, here each time instance is considered to be a possible endpoint to be evaluated. Specific rules are derived and applied to effectively discard non-gesture segments and emit correct gestures.

Thorough experiments have been conducted to validate and compare the three proposed algorithms. Two types of gestures have been tested, one of which involves two arm movements, and the other involves a single hand movement. Experimental results have shown that all three methods achieve high recognition rates, with the last one outperforming the other two.

Earlier versions of this work have appeared in [20–22], and the significant extensions and improvements presented in this paper are:

- Method I presented in [20] can only be applied to isolated gesture recognition. In this work, a new motion detection method is introduced to detect the start time of a gesture and the original framework is modified to work for continuous gesture recognition.
- The first step of the two-phase Early-Decision DP has been derived and a full description has been added in Method III.
- For Method III, the decision rule, which is crucial for determining the end time of a gesture, is modified from its earlier version in [22] and is shown to be more effective.
- Thorough experiments have been conducted to compare all three proposed methods, and intensive analysis and discussion of the experimental results have been included. In previous work, only Method III has been tested on real continuous gesture sequences.
- A new hand gesture database has been collected and tested for Method III, whereas in our previous work, only experiments with arm gestures had been presented.

The rest of the paper is organized as follows. In Section 2, related work will be discussed. Section 3 describes the motion descriptor and gesture representation. Sections 4 and 5 describe Method I, the explicit multi-scale segmentation and recognition, and Methods II and III, which are DP based. Section 6 presents and discusses experimental results. Finally, conclusions are discussed in Section 7.

2. Related work

Several methods have been proposed for continuous gesture segmentation and recognition in the literature. Based on how segmentation and recognition are mutually intertwined, these approaches can be classified into two major categories: separate segmentation and recognition [15,13,14,24], and simultaneous segmentation and recognition [16,17,5,18,32,4,19]. While the first category detects the gesture boundaries by looking into abrupt feature changes and segmentation usually precedes recognition, the latter treats segmentation and recognition as aspects of the same problem and are performed simultaneously. Most methods in both of the two groups are based on various forms of HMM [13,24,16,5,17,19], and DP-based methods, i.e., DTW [15,30,31] and CDP [33,4].

In [15], Kang et al. proposed to detect a candidate cut, which is defined as possible starting or ending points of a gesture, in video games based on three criteria: abnormal velocity, a static gesture, and severe curvature. The segments between those candidate cuts become the possible gesture candidates and are further evaluated using DTW. In [14], Zhu et al. proposed a hand gesture recognition system for HCI, where a hand gesture is spotted based on the temporal length of a moving hand in a video stream. If a hand region appears in the stream longer than L_1 but shorter than L_2 frames, a hand gesture is segmented from the rest of the sequence. A linear resampling technique is then proposed for efficiently recognizing gestures with temporal variations. In [13], the user of a hand gesture recognition system is required to stop for two or three seconds before the gesture starts and after it ends, so that the region between those no-movement areas corresponds to the true gesture. HMM is then employed for gesture recognition based on features which are a combination of weighted location, angle and velocity.

In [24], Liang et al. proposed a gesture recognition system for Taiwanese Sign Language (TWL) by using a DataGlove to capture hand movements. The endpoint of a gesture is first detected by a time-varying parameter (TVP), which is defined as a parameter changing its value over time. Whenever the number of TVP drops below a threshold, the motion is thought to be quasi-stationary, and its corresponding frame is detected as an end point. HMM is then employed for recognizing TWL at the sentence level. Since the requirement that a feature discontinuity must occur is not always satisfied in continuous gesturing, it is questionable whether those methods mentioned above can be generalized to real applications.

For simultaneous segmentation and recognition approaches, gesture boundaries are detected by viewing the recognition scores which are usually obtained by continuously matching the

input to the gesture prototypes in the database. An HMM-based approach that spots dynamic hand gestures in video streams has been proposed in [16]. The output scores of every HMM are continuously observed at every time step using an improved normalized Viterbi algorithm, which allows the output score of an HMM to increase if it describes the momentary input video stream well. Since characteristic peaks in the output scores of the respective models indicate the presence of gestures, a peak finding algorithm is then employed to locate the gesture in the video stream. In [5], Lee et al. proposed a HMM-based threshold model that consists of the states copied from all trained Gesture Models to provide a confirmation mechanism for the provisionally matched gesture patterns. The likelihood of an input sequence is continuously computed for all the Gesture Models and the threshold model. The times that the likelihood of a Gesture Model is greater than that of a threshold model are considered as candidate end points, and are further analyzed to select the best one corresponding to a real gesture. The start point of a gesture can be traced back using the Viterbi algorithm.

In [17], Kim et al. proposed to simultaneously segment and recognize gestures based on forward spotting accumulative HMMs. The endpoints of gestures are first determined by zero crossing points of a competitive differential observation probability, which is defined by the difference of the observation probability between the maximal gesture and the non-gesture. The output scores of all accumulated gesture segments between the start and end points are calculated and a gesture is assigned to the model that has the majority vote of all intermediate recognition results.

In [33,4], Alon et al. proposed a unified framework for spatiotemporal gesture segmentation and recognition. Multiple candidate hand locations are first detected at each frame. Then the feature vectors are matched to the models using Continuous Dynamic Programming (CDP) [34], during which a large number of hand gesture hypotheses can be eliminated using pruning classifiers which are learned from training data. The endpoint of a gesture can be detected if one model gives the lowest matching cost at one time.

In [19], Fang et al. proposed to recognize and segment large-vocabulary continuous sign language based on transition-movement models (TMMs). To reduce the number of TMMs, the transition movements between two adjacent signs are temporally clustered using an improved k-means clustering algorithm. An iterative TMMs training algorithm is presented to automatically segment the sequence into real signs and transition parts, and then train the TMMs and the sign models. The recognition of continuous sign language recognition is achieved by feeding TMMs and new sign models into the Viterbi search algorithm.

In order for gestures to be segmented and recognized from video streams, various models have been employed to characterize gestures. Since human motions are non-rigid movements with high degrees of freedom, a robust model is difficult to obtain. Some efforts have been made to the reconstruction of the human form from image sequences in the belief that such information would be beneficial to interpret the motion. Yacoob et al. [35] proposed to model human motion based on segmenting the human body into five body parts (i.e., arm, torso, thigh, calf, and foot). Each part is tracked individually through images and described by eight motion parameters. In many cases, however, to overcome the difficulties of the reconstruction of human body geometry, various sensors or wearable devices are used to relieve the problem. In [36], hand motion in German Sign Language has been described by hand shape, orientation and location. To obtain those data, the signer is required to wear colored gloves for each hand, with dominant and non-dominant hands marked by different colors. While the glove for the non-dominant hand is one uniform color, the glove for the dominant hand has seven different colors, marking each finger, the palm and the back of the hand. Similarly, Liang et al. [24] represented a gesture in Taiwanese Sign Language using four parameters: posture, position, orientation and motion. A specially designed sensor, i.e., the DataGlove, is required to be worn by the user to track the detailed information about the flexion of 10 finger joints of one hand. A 3D tracker is also used to track the orientation of the hand. In [19], a 48-dimensional vector is used to describe a Chinese sign, including 36 hand shapes, six positions, and six orientations. To obtain those data, two Cybergloves are employed to collect the variation information of hand shape and finger status. Three Pohelmus 3SPACE-position trackers, where two of them are positioned on the wrist of each hand and another is mounted at the signer's back, are used to collect the information of the orientation and position of the hands.

Although intrusive sensors allow for simpler processing, it is very inconvenient for the subject to use in a natural context. Moreover, it is still arguable whether or not such a detailed human geometrical model is necessary when it comes to the issue of interpreting human motion [7]. Bobick et al. [7] proposed to use a set of motion-energy image (MEI), i.e., binary cumulative motion images, and motion-history image (MHI), i.e., scalar-valued images, where more recently moving pixels are brighter, to represent human movement for each view/movement combination. Seven Hu moments are calculated to describe the shapes of MEI and MHI in a translation- and scale-invariant manner. The power of the motion descriptor has been demonstrated on a set of 18 aerobic exercises. In a real-time American Sign Language (ASL) recognition system [37], instead of attempting to obtain a detailed description of hand shape, only hand blobs have been extracted and tracked based on skin color. A sixteen-element feature vector is constructed from each hand's position, change in position between frames, area, angle of axis of least inertia found from the first eigenvector of the blob, length of this eigenvector, and eccentricity of bounding ellipse. High recognition accuracies of ASL at the sentence level proved the effectiveness of the simple hand gesture descriptor. In a framework to automatically extract signs from continuous sign language [38], skin color blobs are first extracted from each frame, where a relational distribution using the edge pixels in the skin blobs is then obtained. A sign can then be represented as a trajectory in a low-dimensional space called Space of Relation Distribution (SoRD), which implicitly captures the shape and motion of the sign.

In another application in which certain types of gestures were to be detected from sports video, due to the low resolution of the player's region, it was almost impossible to recover the player's geometrical body model. Instead, Roh et al. [18] proposed to extract the feature points from the player's silhouette based on curvature scale space (CSS). The posture descriptor was robust even when the silhouette was partially corrupted. Two processes were combined to detect serve gestures in tennis: determination of a posture, and spotting a sequence.

In this work, we will not attempt to obtain a fine description of human body parts, nor do we make use of a geometrical or parts-based body model. Instead, motion is represented by a 3D spatio-temporal surface based on the evolution of a contour over time, described in the next section.

3. Feature extraction and gesture representation

For motion recognition systems, it is important to represent motion at an abstraction level such that different motions could be distinguished by their representations. However, human gestures are non-rigid motions with high degrees of freedom, for

which a general motion descriptor is typically hard to find. If human body geometric information is required for the recognition of motion, then a 2D or 3D model with detailed human body parts information (i.e., the exact locations and the relative locations of head, limbs, torso and so on) is usually generated. Since we believe that motion can be recognized directly without knowing a priori human geometric information, our focus is on choosing features that can describe the dynamic properties of motion to build the model.

Motivated by the observation that motion is composed of a sequence of static shapes whose order is decided by the type of motion performed, a Motion Signature (MS), which is a 3D surface model, has been constructed to accommodate the changes in shapes along spatio-temporal dimensions. To account for variations of motions, a Gesture Model consisting of a set of *mean images* and *variance images* is then constructed using a series of Motion Signatures at multiple scales.

3.1. Motion Signature extraction

It is desirable to parameterize each contour as a shape descriptor that uniquely characterizes the contour. A shape descriptor is a feature vector that preserves important characteristics of a shape, and is ideally invariant to translation, scale (size), and rotation [39]. We chose the 1D distance-to-centroid signal illustrated in Fig. 2 as the shape descriptor. A descriptor with more detailed shape information has been proposed by Belongie et al. [40], where each point on the contour is associated with a *shape context* describing the coarse arrangement of the rest of the shape with respect to the point. In our work, we found that the simpler shape information contained in the 1D signal works very well, and similar motion descriptors have been used in [10,41].

We use a simple motion detection method to segment the subject from the background. Starting with the lower left corner of the foreground region, the border following method proposed by Suzuki et al. [42] is used to extract the contour. Fig. 1(a) illustrates four images from a gesture sequence, (b) shows the results after background subtraction, and (c) shows the final extracted contours.

The length of the 1D descriptor varies with the size of the human, the pose at a given frame, the distance of the subject to the camera, and the intrinsic camera parameters, such as focal length. Even when all these conditions are identical, due to image noise, it is not possible to guarantee exactly the same number of contour points for distinct image frames. We, therefore, normalize the contour size by subsampling the contour signal at equal intervals to a fixed length L, then divide the 1D signal by the largest distance among the points. The resulting shape descriptor D_t at image frame t is given as

$$D_t = [\hat{d}_1, \hat{d}_2, \hat{d}_3, \ldots, \hat{d}_L] \quad 0 \le \hat{d}_i \le 1 \tag{1}$$

where $\hat{d}_i$ is the normalized Euclidean distance between the contour point and the contour centroid. Fig. 2(a) illustrates the contour traversal process, where d_A and d_B are the distances between the centroid and the initial points A and B, respectively. Fig. 2(b) plots the resulting 1D shape descriptor, where $\hat{d}_A$ and $\hat{d}_B$ correspond to the normalized d_A and d_B in (a), respectively.

To represent a sequence of t contours, a matrix M of size $t \times L$ is formulated from all poses during the motion period:

$$M = [D_t, D_{t-1}, \ldots, D_1]_L^{\mathrm{T}} \tag{2}$$

The shape descriptors are ordered temporally such that the first frame lies at the bottom row of the matrix.

We called M a Motion Signature, which describes a motion as a 3D spatio-temporal surface. This surface can be visualized by adding a time axis orthogonal to the 2D spatial plane shown in Fig. 2(b), and then stacking each of the shape descriptors sequentially. The 3D surface can also be visualized as a 2D graylevel image with the x axis denoting the contour points and the y axis as time (i.e., frame number). Fig. 3 shows both the 3D and 2D representations of M for three gestures, i.e., "wave left hand","wave right hand" and "wave two hands". It can be observed that the Motion Signature captures the dynamic properties of each motion and exhibits different patterns for each distinct gesture.

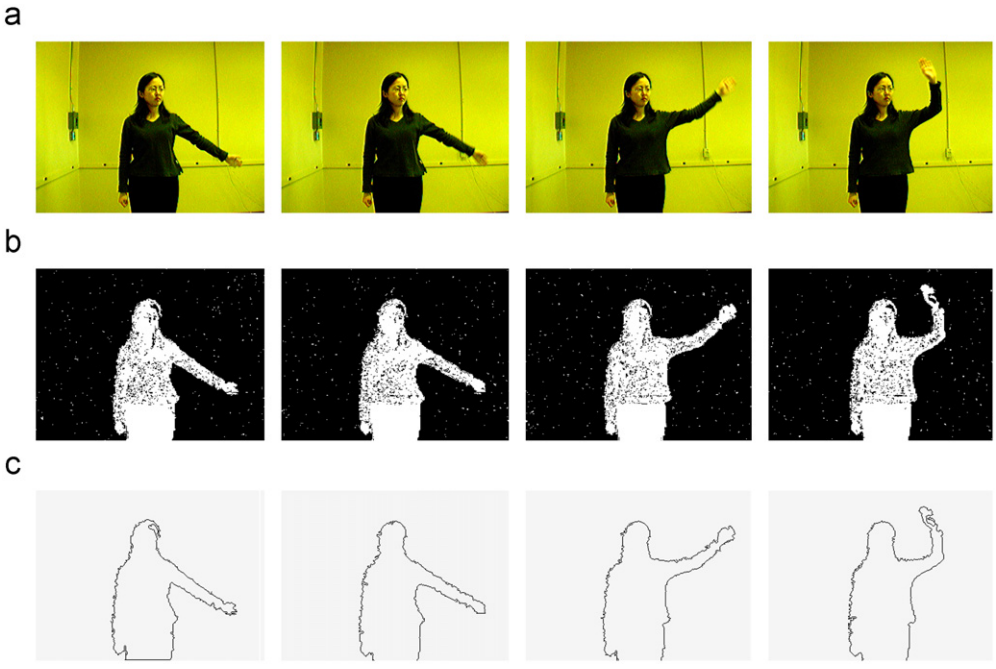

Fig. 1. Background subtraction and contour extraction: (a) original images, (b) images after background subtraction, and (c) extracted contours.

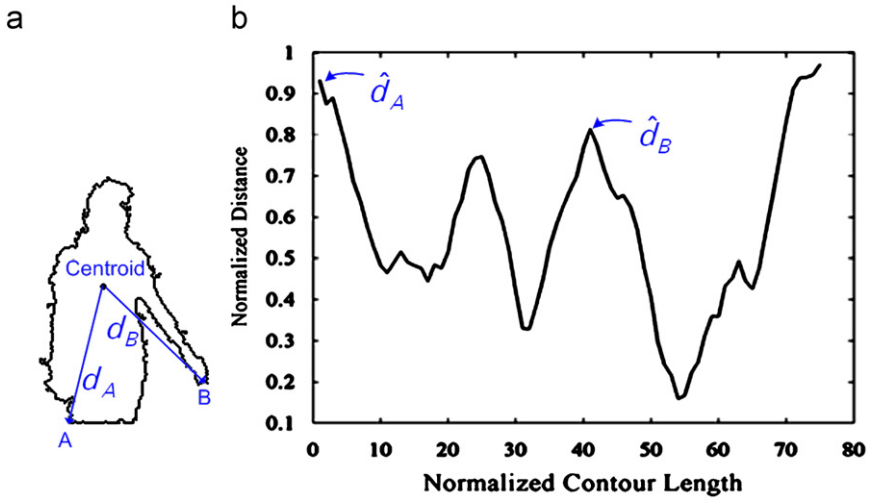

Fig. 2. Contour-based shape descriptor: (a) contour traverse and (b) normalized 1D distance signal.

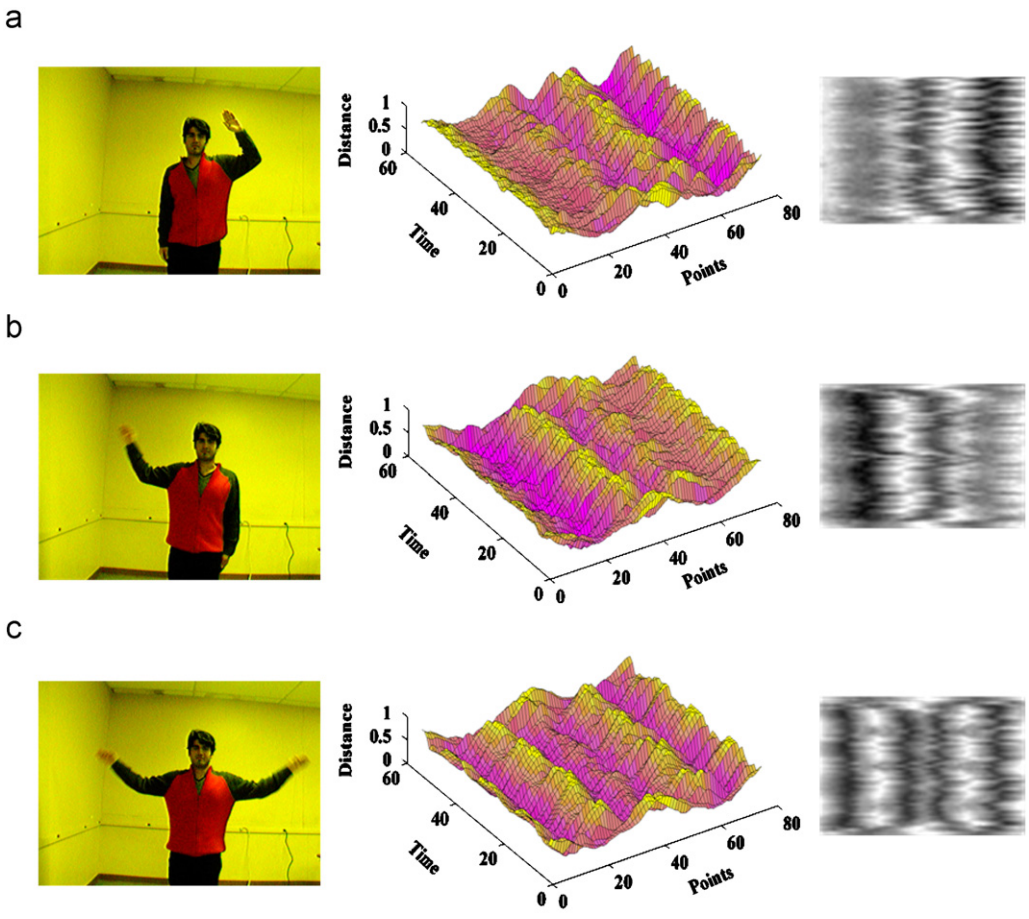

Fig. 3. 3D and 2D representation of Motion Signature for gesture: (a) wave left hand, (b) wave right hand and (c) wave two hands.

3.2. Gesture Model

While different motions will have significantly different Motion Signatures, distinct repetitions of a single gesture will also exhibit some spatial and temporal variations. Among the factors contributing to this variation, motion speed (i.e., scale) is the most influential. An example is given in Fig. 4, which shows the Motion Signatures of multiple instances of a single gesture, "raise left hand", performed at arbitrary speeds. Fig. 4(a) shows the Motion Signature performed during the training stage at a moderate speed. The images in (b) are gestures with faster speeds, and (c) with slower speeds. It can be seen that the Motion Signatures appear to be either compressed when the motion is fast, or dilated when the motion is slow.

To estimate the motion scale, we construct a Gesture Model consisting of a set of Motion Signatures at various scales. A

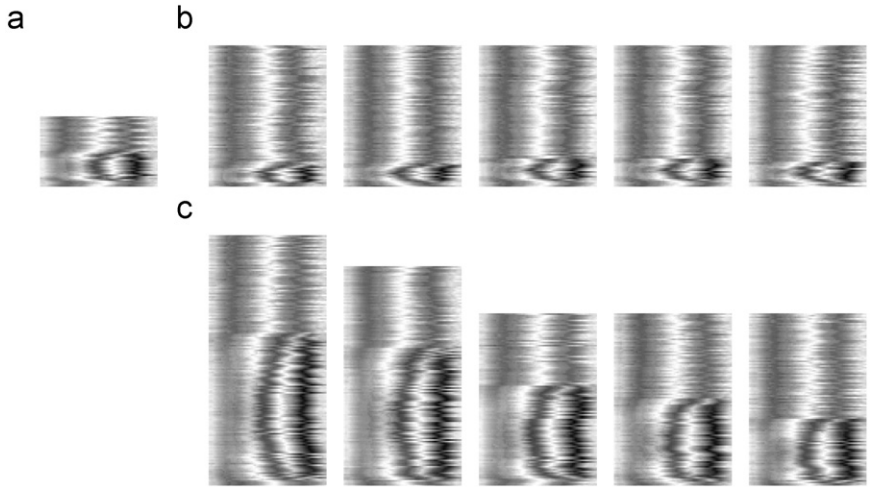

Fig. 4. Gesture "raise left hand" shows large temporal scale difference: (a) training gesture at a moderate speed, (b) faster gestures and (c) slower gestures.

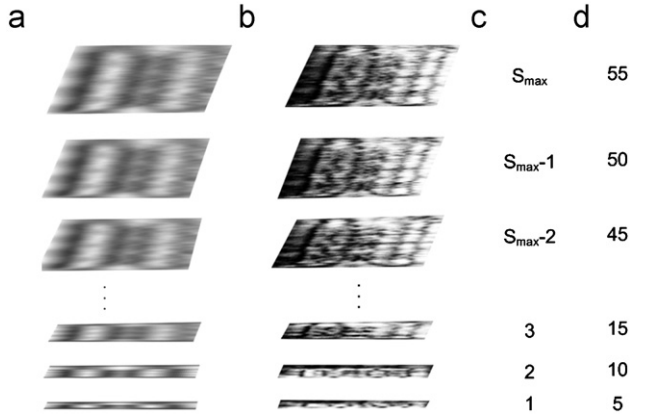

Fig. 5. Gesture Model containing multiple scales for gesture "wave two hands": (a) mean images, (b) variance images, (c) scale and (d) number of frames.

number of image sequences are required for each gesture at each scale, and it would be labor intensive to collect Motion Signatures for all possible scales during training. As an alternative, only a single moderate scale is collected directly for each gesture, while the others are interpolated from this data. We found that if the actor does not change speed abruptly during one motion period (thereby combining multiple scales in one gesture), then this strategy produces very accurate results. To build a Gesture Model, the actor is required to perform a gesture repeatedly at a single moderate speed in the training process. The speed of each training gesture is not required to be strictly the same. Indeed, some variations are beneficial to the building of Gesture Models.

We assume that the scale s_{max} of the training gesture is the slowest scale that will be encountered. Given that each instance of the training motions lasts t frames and that the interval between two successive scales is set to ε frames, the total number of scales will be t/ε, which is equal to s_{max}. The Motion Signature of the training data at scale s is produced using bilinear interpolation with the function ϕ:

$$M_s = \phi(M_{s_{max}}, s, \varepsilon) \quad 1 \le s \le s_{max} \tag{3}$$

With the training set of N occurrences of each gesture, the *mean image* and *variance image* are obtained using Eqs. (4) and (5), respectively,

$$\mu_s(x,y) = \frac{1}{N}\sum_{j=1}^{N} M_{js}(x,y) \tag{4}$$

$$\sigma_s(x,y) = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}(M_{js}(x,y)-\mu_s(x,y))^2} \tag{5}$$

Here, $M_{js}(x,y)$, $\mu_s(x,y)$ and $\sigma_s(x,y)$ refer to the jth training surface image, mean image and variance image at scale s, respectively. The mean and variance images are obtained by calculating the mean and the standard deviation of each point on the Motion Signature. These two images capture reliable information of the motion while excluding the details that vary across training trials. Fig. 5 shows an example of the Gesture Model for the motion "wave two hands". The 1st and 2nd columns show the mean and equalized variance images at multiple scales, with the bottom images indicating smaller scales. The 3rd and 4th columns indicate the scale and the number of frames in each image, respectively.

4. Explicit multi-scale segmentation and recognition

In this section, an approach is described that explicitly segments gestures from a sequence based on Gesture Models described in the previous section. Gesture segmentation and recognition are considered here to be two aspects of the same problem. The process of locating the end time of a gesture is achieved by matching motion segments to the predefined prototypes in the database in a multi-scale manner, which actually is a recognition process. Therefore, once the endpoint of a gesture has been determined, the gesture can then be recognized automatically.

4.1. Starting point detection and recognition

Given a stream of gesture signal, since a motion cannot be infinitely slow, it is reasonable to set an observation point at time t_p such that at least one gesture is included backwards from this time. Once a gesture has been recognized, the end time of that gesture then becomes the start time of the next segment, and another observation point is then set such that the length of the segment remains the same. In this way, gestures are continuously recognized in a video stream.

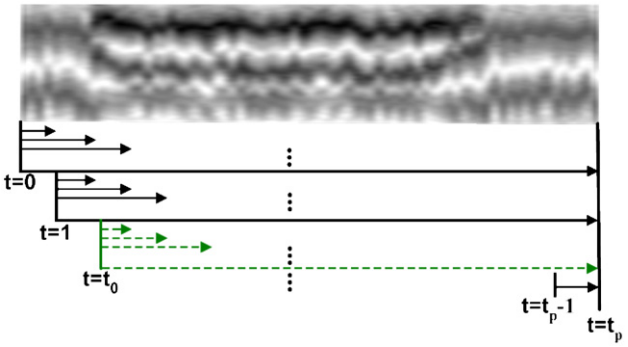

Fig. 6. Motion segments with varying start points and scales. The arrow lines represent possible gesture segments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

If every time instance is allowed to be a starting point and gestures are time-varying (i.e., a given gesture may occur at a variety of speeds), then the total number of gesture candidates will be: $\sum_{i=1}^{t_p} i$, as shown in Fig. 6. Obviously, matching all of these candidates to the Gesture Models is computational expensive, especially for a large database. More efficiently, we employ a motion detection method to find the starting point of a gesture. For that purpose, a non-Gesture Model "stand still" representing a subject standing still without any motion, has been created using the same method as that of the Gesture Models. Starting from the first frame of the input segment, each frame is matched to the non-Gesture Model and the matching score is compared to a predefined threshold. Once the score is above the threshold, it can be concluded that a motion occurs and a starting point is set. In Fig. 6, suppose that t_0 is the starting point found, then the green dotted arrow lines are the motion segments left after the motion detection process. It can be seen that only a very small percentage of the motion segments remain. A multi-scale matching scheme is now employed to find the best match among the motion segments. Whenever a motion segment is shorter than the maximum scale s_{max} of a given Gesture Model, a similarity measure is evaluated between these motions at the corresponding scale; otherwise, the motion is compressed to the maximum scale and the comparison is conducted at the maximum scale. This process is illustrated in the following equations:

$$M^*_{kt_0s} = \arg\max_{k,s} \begin{cases} f(M_{t_0s}, G_{ks}) & 1 \le s \le s_{max} \\ f(M'_{t_0s}, G_{ks_{max}}) & s > s_{max} \end{cases} \quad (6)$$

where $M'_{t_0s} = \phi(M_{t_0s}, s_{max}, \varepsilon)$ and $1 \le k \le K$.

In the above, K is the total number of motion patterns in the database. M_{t_0s} is a test Motion Signature from starting point t_0 at scale s, and G_{ks} is the Gesture Model k at scale s. M'_{t_0s} is interpolated from M_{t_0s} using the function ϕ defined in Eq. (3). The gesture and correct scale are assigned to the Gesture Model with the maximum value found in Eq. (6). The function f is used to measure the similarity between two motions. The choice of f is discussed in the next section.

4.2. Similarity measurements

Two similarity measurements, i.e., Correlation and Mutual Information [43,44] are considered for function f.

The Correlation function f_C is defined as

$$f_C = \sum_{i=1}^{N_i} \frac{|M_s(i) - \mu_s(i)|}{\sigma_s(i)} \quad (7)$$

where μ_s, σ_s are the mean and variance images of a Gesture Model and M_s is a test motion at scale s. N_i is the total number of pixels in the images. The input is classified as the gesture and scale with the minimum value of Eq. (7).

While Correlation considers spatial information, Mutual Information explores the statistical dependence of two patterns. The Mutual Information (MI) between two images is calculated as

$$f_M = \sum_l \sum_k p(l,k) \times \log \frac{p(l,k)}{p_{M_s}(l) p_{\mu_s}(k)} \quad (8)$$

where $p_{M_s}(l)$ and $p_{\mu_s}(k)$ are the marginal densities of a motion segment and the mean image of a Gesture Model at scale s, respectively, and $p(l, k)$ is the joint intensity. A detailed derivation of f_M can be found in [20].

It will be shown that both Correlation and MI work very well, with Mutual Information performing slightly better in terms of recognition rate. This is because MI combines both spatial and structural information of images to measure similarity, while Correlation only uses spatial information. However, the calculation of Correlation is much simpler and more efficient than that of MI, with a complexity of $O(N)$ (N is the total number of pixels in an image) for Correlation vs. $O(NM^2)$ (M is the total number of bins of the density) for MI. Therefore, the choice of Correlation or MI is a trade-off between accuracy and efficiency.

5. Continuous gesture recognition using Dynamic Programming

In this section, two approaches are presented that recognize continuous gestures by making use of Dynamic Programming (DP). DP is a technique that can effectively match time sequences when a time scale variation exists by recursively making optimal decisions at any intermediate stage. This makes it an ideal candidate for matching gesture signals, where a nonlinear alignment in time is essential. However, because of the endpoint localization issue, DP cannot be applied directly to recognize gestures in a continuous stream. Dynamic Time Warping (DTW), which is an application of DP and has been widely used in gesture recognition, has unfortunately only been applied to isolated gesture recognition [27,28,12].

In Section 5.1, DTW will be introduced, followed by an approach that extends DTW for continuous gesture segmentation and recognition. In Section 5.2, an approach based on the derivation of DP will be presented.

5.1. Endpoint localization using Dynamic Time Warping

The principle of DP states that the final optimal path relies only on the initial state and the intermediate optimal decisions made along the path. The mechanism of DTW of matching two signals is to find the global optimal path by recursively accumulating locally optimal paths. Fig. 7 shows the nonlinear mapping between a reference gesture signal $M_{t'}$ and a test one M_t. The shaded region represents the global parallelogram constraint of DTW, which excludes certain regions of the DP plane that the warping path can lie within. The best time warp will minimize the accumulated distance along the path through the grid from (1, 1) to (T,T'). When the local range of the path in the vicinity of the point (t,t') is restricted to its immediate three neighbors (i.e., local constraint type I shown in the upright corner in Fig. 7), DTW can be formulated as

$$R_{t,t'} = r_{t,t'} + \min(R_{t,t'-1}, R_{t-1,t'}, R_{t-1,t'-1}) \quad (9)$$

and the initial conditions are

$$R_{1,1} = r_{1,1}$$

$$R_{t,1} = r_{t,1} + R_{t-1,1}$$

$$R_{1,t'} = r_{1,t'} + R_{1,t'-1} \tag{10}$$

In Eq. (9), $R_{t,t'}$ is the partial accumulated distance, and $r_{t,t'}$ measures the distance of gesture signals at two temporal instances:

$$r_{t,t'} = \sum_{j=1}^{L} \frac{|M(t,j) - \mu(t',j)|}{\sigma(t',j)} \tag{11}$$

where $M(t,j)$ is the input gesture at time t, and $\mu(t',j)$ and $\sigma(t',j)$ are the mean and variance image of the Gesture Models at time t'. L is the normalized length of the contour. The recursion of Eq. (9) constitutes the forward stage of DTW, where $R_{t,t'}$ is computed within the shaded region until the endpoint (T,T') is reached and its value $R_{T,T'}$ is calculated. The optimum path can be obtained by tracing back the grid through (T,T') to (1, 1).

Note that in the case of isolated gesture recognition, the endpoints have been precisely determined in advance for both the reference and test signals. Therefore, a single warping path starting at a fixed starting point and ending at a fixed termination point can be found by DTW for each Gesture Model. Given a sequence with multiple gestures performed in an arbitrary order, it is obvious that DTW cannot be applied directly since the endpoints of each gesture are unknown.

An extension of DTW, i.e., Level-Building DTW, for the application of connected word recognition has been proposed by Myers and Rabiner [29]. The spoken sequence is warped to a successive concatenation of reference patterns in a level-by-level manner. At each level, the end of each reference pattern and the upper and the lower global constraint lines form a region in which the dynamic warping is conducted. The accumulated distance scores and best reference patterns are retained for each level and serve as the initial conditions for the next level until the last level and the end of the sequence are reached. The best warping path can be traced back from the last level and the matching reference patterns can thus be obtained. Recently, an enhanced version of the Level-Building DTW has been proposed to particularly handle the movement epenthesis (me) problem in continuous sign language recognition [30,31], where a "me" label is inserted between actual signs whenever a segment is determined not from a real sign.

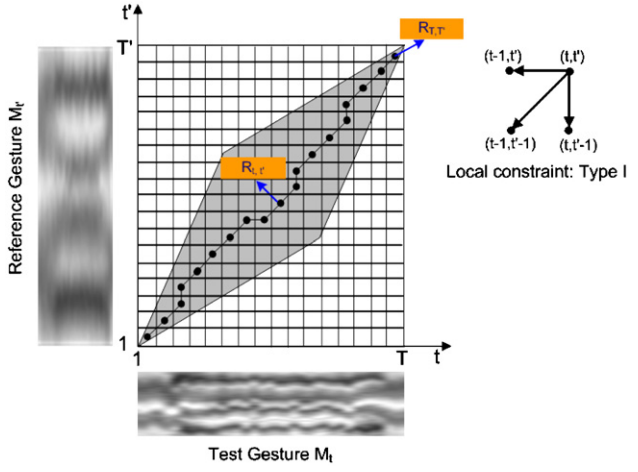

Fig. 7. Dynamic Time Warping of a test gesture to a reference gesture and the local constraint of Type I.

It should be pointed out that the Level-Building DTW is a deferred-decision algorithm, in the sense that all the words or signs have to be spoken or performed before the algorithm can be carried out. In gesture recognition, however, an early-decision system is generally preferred since it is desirable to determine the meaning of a gesture immediately after it has been performed, without having to wait until all remaining gestures have been executed. Furthermore, the number of levels in the Level-Building DTW is determined by the number of words spoken, which should be known in advance. Although this requirement can be relaxed in the case of unknown lengths by simply comparing the accumulated distance scores associated with each of the potential lengths, it is difficult to decide how many varied levels should be tested when the lengths of the gesture sequences are totally random.

A two-phase, early-decision recognition algorithm that makes use of DTW for endpoint estimation is proposed. For that purpose, a Compound Gesture Model, which sequentially concatenates two Motion Signatures taken from possibly different Gesture Models at the maximum scale is constructed. Without any assumptions on the temporal sequence of gesture occurrences, a gesture could be succeeded by any other gesture, including itself. We call the first gesture the *head* gesture and the second the *tail* gesture. To account for the occurrence, where a gesture is followed by a non-gesture (i.e., "stand still") we treat "stand still" as a normal gesture, which could appear to be either the *head* or *tail* gesture in a Compound Gesture Model. Fig. 8 shows some mean images of Compound Gesture Models when the *head* gesture (in the bottom half of the image) is "wave right hand", and the *tail* gestures are "wave right hand", "wave two hands", "raise right hand","left down right up" and "stand still". It can be seen that the lengths of the Compound Gesture Models vary accordingly depending on the concatenated Gesture Models.

In the Compound Gesture Models, we know exactly the start and end times of the *head* gesture, and this knowledge can be used to estimate the temporal endpoints of the input gesture. Looking backwards in time over a fixed period in the input sequence, it is possible that the input sequence contains two sequential gestures, the first of which is complete, and the second of which is as yet incomplete. By warping the test sequence to the closest Compound Gesture Model, the endpoints of the first gesture can thus be estimated. Fig. 9 illustrates this idea. Suppose that $P_{T,T'}$ is the warping path between the Compound Gesture Model and the test segment, and t'_0 and t'_1 are the start and end

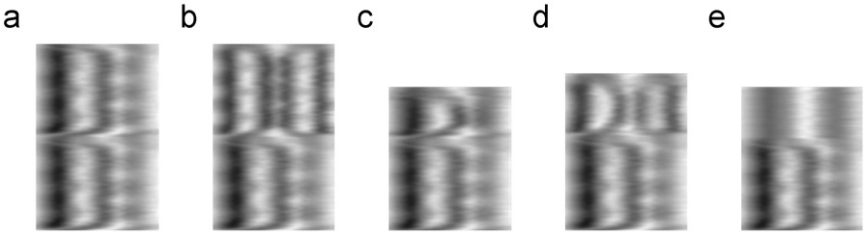

Fig. 8. Compound Gesture Models for "wave right hand" given that the subsequent gestures are: (a) "wave right hand", (b) "wave two hands", (c) "raise right hand", (d) "left down right up" and (e) "stand still".

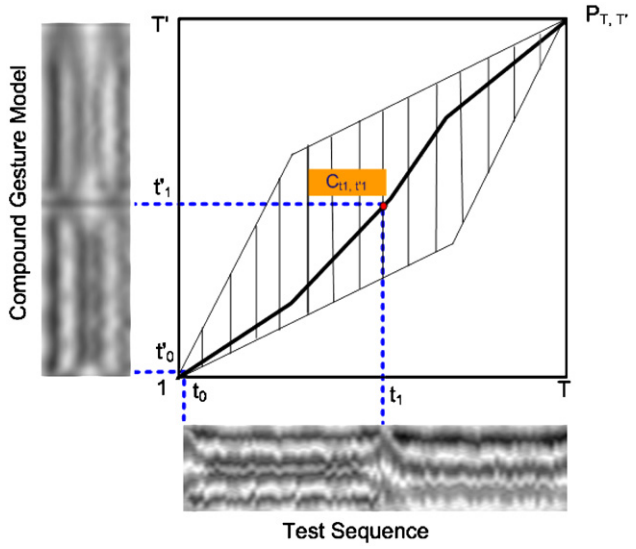

Fig. 9. Temporal endpoint estimation by warping a test sequence to a Compound Gesture Model using DTW.

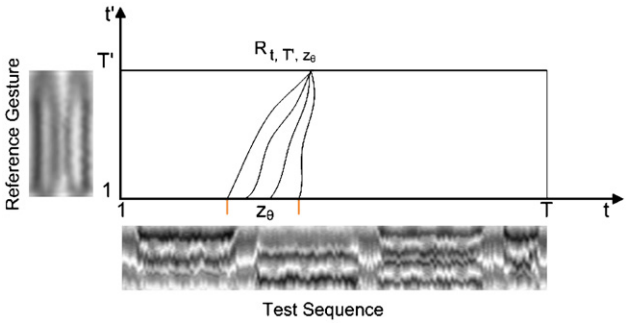

Fig. 10. Dynamic Programming of a test gesture to a reference gesture.

times of the *head* gesture of the model. From the intersection point C_{t_1,t'_1}, the end time of the test gesture t_1 is to be found. The start time t_0 can also be obtained in the same way. Notice that t'_0 is not the first frame of the Compound Gesture Model, but rather is the second frame. This is because the first frame of the Compound Gesture Model is always aligned to the first frame of the test sequence due to the endpoint constraint of DTW. By setting t'_0 to the second frame, we bypass the problem that the start time of the test gesture is fixed to be the first frame.

The two-phase recognition process is as follows. The temporal endpoint estimation method described above serves as an initial step to segment the gesture from the rest of the sequence. Warping two very different signals will result in a large accumulated distance value, so those Compound Gesture Models with the smallest DTW distances are considered to be the most similar to the test motion segment. In the second phase, only those Compound Gesture Models with the smallest DTW distances are further segmented around the endpoints and tested against the Gesture Models using the multi-scale recognition method described in Eq. (6). More precisely, only a total of 15 segments, whose contexts vary slightly according to the start times (i.e., $t_0 \pm \varepsilon$, t_0) and scales (i.e., $s \pm 1$, $s \pm 2$, and s), are tested for each Gesture Model using Eq. (6). Again, either Correlation or Mutual Information can be used in matching.

Note that instead of matching all Gesture Models at all possible temporal scales to find the end time of a gesture as Method I described in the previous section, only a very limited number of scales remain to be examined by Method II. Therefore, it is expected that the accuracy and efficiency will be improved.

5.2. Early-decision Dynamic Programming

When DP is applied directly to the application of continuous gesture recognition, instead of one unique optimum path to be found starting and ending at fixed points for each model, there will be multiple paths associated with each gesture prototype in the DP space. Fig. 10 shows a gesture sequence of length T warped to a reference model of length T'. All the warping paths should end on the line $t' = T'$, and because of the unknown starting point, each ending point (t,T') could originate from many possible starting points z_θ. The number of allowed starting points depends on the local and global constraints of DP, which limits the DP warping areas such that some unreasonable paths could be excluded in the first place. Suppose there are N_θ starting points allowed for each ending point. Then, N_θ is usually much smaller than T and has to be in the range of $1 \le N_\theta \le T$. R_{t,T',z_θ} represents the accumulated distance value for the path originating from point $(z_\theta,1)$ and terminating at (t,T').

Since an early-decision system is generally preferred, R_{t,T',z_θ} has to be evaluated each time a new feature vector is obtained, and a decision is made as to whether the end of a correct gesture has been reached. This can be achieved by breaking down the DP algorithm into a two-step procedure: first, for all prototypes, the best path ending at (t,T') is found among the N_θ allowable paths, and the starting point z_θ of the best path and the minimum value of R_{t,T',z_θ} are recorded; second, at each time instance, a conclusion whether a correct gesture is present or not is to be drawn based on the data obtained from the first step. Although the value of R_{t,T',z_θ} usually comes close to a local minimum as a correct gesture approaches its end, it is not trivial to determine when a gesture occurs depending solely on R_{t,T',z_θ}. It will be shown later that a method has been developed that can reliably detect a gesture by combining and analyzing all the information available.

A real advantage of the two-step procedure lies in that the principle of optimality holds for the first step, which significantly simplifies the calculation of multiple warping paths [25]. Following [25] for the derivation of simplification, the first step of minimizing the N_θ starting points can be expressed by

$$R_{t,T'} = \min_{1 \le \theta \le N_\theta} R_{t,T',z_\theta} \tag{12}$$

$$\overline{z}_\theta = \arg \min_{1 \le \theta \le N_\theta} R_{t,T',z_\theta} \tag{13}$$

If a type I local constraint as shown in Fig. 7 is applied, by inserting Eq. (9) into Eq. (12), it becomes

$$R_{t,T'} = \min_{1 \le \theta \le N_\theta} [r_{t,T'} + \min(R_{t,T'-1,z_\theta}, R_{t-1,T',z_\theta}, R_{t-1,T'-1,z_\theta})] \tag{14}$$

Taking a close look at Eq. (14), it can be seen that $r_{t,T'}$ is a constant, and therefore, can be safely moved out of the minimization function. Meanwhile, since the minimization functions are independent, their execution order is actually exchangeable. Thus

$$R_{t,T'} = r_{t,T'} + \min\left[\min_{1 \le \theta \le N_\theta} R_{t,T'-1,z_\theta}, \min_{1 \le \theta \le N_\theta} R_{t-1,T',z_\theta}, \min_{1 \le \theta \le N_\theta} R_{t-1,T'-1,z_\theta}\right] \tag{15}$$

Defining the following quantities:

$$\overline{R}_{t,t'} \equiv \min_{1 \le \theta \le N_\theta} R_{t,t',z_\theta} \tag{16}$$

$$\overline{z}_{t,t'} \equiv \arg \min_{1 \le \theta \le N_\theta} R_{t,t',z_\theta} \tag{17}$$

Then Eq. (15) becomes

$$\overline{R}_{t,t'} = r_{t,t'} + \min[\overline{R}_{t,t'-1}, \overline{R}_{t-1,t'}, \overline{R}_{t-1,t'-1}]$$

$$\overline{z}_{t,t'} = \text{argmin}[\overline{R}_{t,t'-1}, \overline{R}_{t-1,t'}, \overline{R}_{t-1,t'-1}] \quad (18)$$

The initialization conditions are

$$\overline{R}_{t,1} = r_{t,1}$$

$$\overline{z}_{t,1} = t \quad (19)$$

The initial value of $\overline{z}_{t,1}$ is set to t, which indicates that every time instance could be a potential starting point of a path. Eq. (18) states that the minimization over the starting point actually can be done recursively, which is much more computationally efficient since $\overline{R}_{t,t'}$ is only calculated once for each point in the DP space. Note that the type I local constraint used in the above derivation can be replaced by any other type of local constraint, such as CDP in [34].

Given that $\overline{R}_{t,t'}$ is calculated per prototype per time instance, ideally, a gesture can be segmented and recognized as long as the accumulated distance value for one pattern falls below a predefined threshold during the warping process. Usually, a threshold for each pattern has to be defined in advance based on training data. However, due to the large spatial and temporal variations among gestures, it is very difficult to find appropriate thresholds based on limited training data. Moreover, since threshold values vary among different gesture classes, a decision will be hard to make when more than one gesture have accumulated distance values below their own thresholds at roughly the same time. Therefore, segmenting continuous gestures by thresholding $\overline{R}_{t,t'}$ usually will not give satisfactory results.

In our approach, the endpoint detection will not depend on thresholds learned from each individual class. Only one uniform threshold Γ is used to rule out those warping paths with sufficiently large accumulated distances, and correct gestures are determined by incorporating all the information available.

At each time t, $\overline{R}_{t,t'}$ is computed for every prototype. Whenever $\overline{R}_{t,t'}$ is smaller than Γ, the time when the value is observed, the gesture class, the start time (which can be traced back through the warping path), together with $\overline{R}_{t,t'}$ are put into the gesture candidate list α. The first candidate that enters the list is marked as the "best" candidate, and for all the other candidates four rules are applied:

1. If a gesture candidate has a smaller accumulated distance value than the "best" candidate, and its start time is close to the start time of the "best" candidate, then this candidate becomes the new "best" candidate, and the previous "best" one is deleted from the list.
2. If a gesture candidate has a larger accumulated distance value than the "best" candidate, and its start time is close to the start time of the best candidate, then delete the gesture candidate.
3. If a gesture candidate has a larger distance value than the "best" candidate, but its start time is later than or close to the end time of the "best" candidate, then the "best" candidate is pronounced as a real gesture, and the current gesture is marked as the new "best" candidate.
4. If a gesture has a larger distance value than the "best" candidate, and its start time is later than the start time of the "best" candidate, but earlier than the end time of the "best" candidate, then delete the gesture candidate.

When all gesture candidates in the list have been checked, we proceed to the next frame and the above process is repeated, until the end of the sequence is reached.

From the above description it can be seen that a gesture can be recognized immediately after it has been enacted, and therefore, an early-decision recognition system is achieved.

6. Experiments

Thorough experiments have been conducted to evaluate all three methods proposed. Two types of gesture have been used, one of which involves two arm movements with the whole upper body extracted for contour extraction. The other includes a single hand movement with the hand's contour used as the feature vector. The first type of gesture were captured with a Point Grey FireFly camera using an image resolution of 320×240, at a frame rate of 15 fps, while the second type was acquired with a Point Grey DragonFly camera with the same resolution and acquisition speed as the first one.

6.1. Single gesture tests

The experiment was started from a relatively simple situation, where each testing sequence depicted a subject executing a single gesture. The goal was to evaluate the descriptive power of the gesture representation and the effectiveness of the segmentation and recognition algorithms. Only the first method, which explicitly segments and recognizes gestures in a multi-scale manner, was applied in this experiment. Note that although the testing was for the single gesture case, the start and end times of a gesture in the sequence were actually unknown, which made it significantly different from isolated gesture recognition. In IGR, the endpoints of a gesture are known a priori, so no segmentation is necessary.

In total, eight gestures performed by five subjects have been collected. Gesture nos. 1–3, namely "wave right hand", "wave left hand" and "wave two hands", were periodic gestures which contain repeated waving motions, while the remaining gesture nos. 4–8, "raise right hand", "raise left hand", "raise two hands", "left down right up" and "left up right down" were non-periodic. Since periodic gestures lasted longer than the others, the values of their maximum scale s_{max} were larger. Given that the scale interval ε was set to 5, s_{max} of the training data for the first three gestures was set to 11 (55 frames). For gesture nos. 4–6, s_{max} was set to 6 (30 frames) and s_{max} was set to 35 for the last two gestures. The normalized length L of the shape descriptor was 75. Fig. 11 shows some sample frames from the eight gestures performed by five subjects.

Each gesture was trained using 30 instances performed at similar rates by three subjects. During the training process, the gestures were manually segmented from the sequence to build the Gesture Models. For testing, another 60 instances from two different subjects were collected. The testing data were divided into two data sets. In data set 1, each gesture included 40 sequences that have similar temporal scales to that of the training data, i.e., at normal speed. The length for each sequence was 75 frames. Data set 2 was composed of another 20 sequences per gesture, where the subjects were asked to intentionally move very fast or very slow such that the temporal scales were significantly different from the training data scale. The lengths of these sequences varied from 100 to 200 frames. The purpose of the first single scale test was to provide a general idea of the performance of the approach under normal conditions, while the second multiple scale test evaluated the robustness of the approach under arbitrary conditions. Both similarity measurements, i.e., Correlation and Mutual Information were implemented.

Table 1 shows the recognition rate per gesture and the average recognition rate for all gestures using Correlation and MI. Not surprisingly, Method I achieved a higher recognition rate for data set 1 than data set 2 with either Correlation or MI, although the rate was still above 86% in the second case, which demonstrates

Fig. 11. Sample frames from eight gestures: "wave right hand", "wave left hand", "wave two hands", "raise right hand", "raise left hand", "raise two hands", "left down right up", and "left up right down".

that Method I is robust to varied temporal scales. Furthermore, the subjects used for testing were not the same as those in the training data, which demonstrates that our gesture representation works very well for different subjects.

As far as the two similarity measurements are concerned, Correlation performed better than MI in the first data set, with recognition rates at 99.4% vs. 89.7%, respectively. Similar rates were observed in data set 2, at 86.3% vs. 86.9%, respectively. It is

Table 1
Recognition rates for single gesture test using Method I.

	1	2	3	4	5	6	7	8	Average
Correlation I	100.0	95.0	100.0	100.0	100.0	100.0	100.0	100.0	99.4
Correlation II	100.0	90.0	90.0	85.0	100.0	100.0	55.0	70.0	86.3
MI I	100.0	95.0	100.0	65.0	90.0	100.0	90.0	77.5	89.7
MI II	100.0	95.0	100.0	75.0	100.0	100.0	70.0	55.0	86.9

Table 2
Average confusion matrix for single gesture test using Correlation on data set 2.

No.	1	2	3	4	5	6	7	8
1	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	90.0	0.0	0.0	10.0	0.0	0.0	0.0
3	0.0	0.0	90.0	0.0	0.0	0.0	5.0	5.0
4	15.0	0.0	0.0	85.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
7	0.0	0.0	0.0	0.0	45.0	0.0	55.0	0.0
8	0.0	0.0	0.0	0.0	30.0	0.0	0.0	70.0

interesting to see that Correlation achieved almost 100% recognition rate when there was little change in temporal scale, but its performance dropped quickly as the changes increased. MI, on the other hand, had a relatively stable performance in either situation.

From Table 1, it can also be found that some gestures were more easily confused with others. In general, gesture nos. 4, 7 and 8, namely, "raise right hand", "left down right up" and "left up right down", were more frequently recognized as other gestures. Table 2 is the confusion matrix showing the average recognition rates for the test using Correlation on data set 2. It can be seen that "raise right hand" sometimes was recognized as "wave right hand". Further inspection of those gestures found that this happened when "raise right hand" moved very slowly and its Motion Signature distorted significantly from its model. The same thing happened to gesture "left down right up" and "left up right down". Usually, the method failed if the Motion Signature of the gesture was dramatically different from that of the models, and recognition from the model then became very hard.

6.2. Multiple gesture tests

In this experiment, all three methods were tested thoroughly for continuous gesture recognition. A total of 80 video clips performed by three subjects, one of them not among the training subjects, were collected. For each video clip, the subject was asked to arbitrarily perform any eight gestures continuously in random order, without any pauses between gestures. Each sequence could contain all eight gesture types, or have some gestures repeated. In total, there were 640 gestures for testing. All 80 video clips were manually segmented and annotated as the ground truth. Fig. 12 shows such a sequence spanning 400 frames. The vertical bars among adjacent gestures were added for clear viewing only. Although some pauses ("stand still") between gestures can be observed from Fig. 12, those pauses were not made intentionally for easy segmentation. In the next section, it can be seen that there were no such pauses at all for hand gesture sequences (shown in Fig. 14).

For the automatic segmentation and recognition process, the total number of gestures in the sequence is actually unknown, therefore, the length of the output could be longer or shorter than its true length (i.e., 8 here). To evaluate the results, the criterion used in continuous speech recognition was employed, where the recognition rate was based on three error types: *Substitution*, where an incorrect gesture was substituted for the correct one; *Deletion*, where a correct gesture was omitted in the recognized sequence; and *Insertion*, where an extra gesture was added in the recognized sequence.

The recognition rate was then calculated as [37]

$$Rec.\ Rate = 100\% \times \left(1 - \frac{Subs + Dels + Ins}{No.\ of\ gestures}\right) \quad (20)$$

Table 3 shows the results for all 640 gestures from all the 80 clips of all the three methods, among which the third method, i.e., early-decision DP, achieved the best recognition rate of 96.4%. This showed that the proposed DP-based approach effectively segmented and recognized gestures in a continuous stream.

The second method, which used DTW to localize the endpoints, obtained a recognition rate of 90.8% and 91.4%, when Correlation and MI were used, respectively. The recognition rates for Method I were a little bit lower than Method II, which were 86.9% for Correlation and 88.9% for MI, respectively. The major difference between Methods I and II was that the former found the endpoints of a gesture by motion detection and multi-scale searching, while the later used Dynamic Time Warping of Compound Gesture Models. Since the rough position of the endpoints of a gesture had been estimated before accurately localizing for Method II, it is not surprising that Method II gave better results than Method I, which searched through all possible temporal scales to localize the endpoints of a gesture.

For both Methods I and II, MI gave slightly better results than Correlation. This conclusion seems to be contrary to that of the first experiment, where Correlation performed better than MI. However, if the results were compared to that of the first experiment, it can be found that MI generated fairly stable results across experiments, approximately between 86.9% and 91.4%. Correlation, on the other hand, can perform extremely well in some situations, like the 99.4% rate obtained from the first experiment on data set 1, but in general also resulted in an

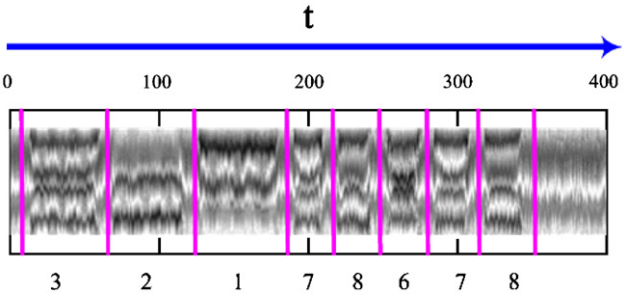

Fig. 12. A continuous gesture sequence containing eight gestures. From left to right: 3 ("wave two hands"), 2 ("wave left hand"), 1 ("wave right hand"), 7 ("left down right up"), 8 ("left up right down"), 6 ("raise two hands"), 7, 8.

Table 3
Recognition rates for multiple gesture test.

	Subs	Dels	Ins	Rec. Rate (%) (total no. 640)
Method I: Corr.	54	25	5	86.9
Method I: MI	19	2	50	88.9
Method II: Corr.	45	14	0	90.8
Method II: MI	41	11	3	91.4
Method III	13	6	4	96.4

average recognition rate between 86.3% and 90.8%. So it can be concluded that the choice of Correlation or MI makes only a small difference in the outcome of the experiments.

Table 3 also shows that, for Method I, as far as the three error types are concerned, *Substitution* error occurred much more frequently than *Deletion* and *Insertion* when Correlation was used, i.e., 54 Subs vs. 25 Dels and 5 Ins; while when MI was applied, *Insertion* error was dominant, i.e., 50 Ins vs. 19 Subs and 2 Dels. This further demonstrated that Correlation and MI worked differently in terms of similarity measurement, i.e., while Correlation only used spatial information of images to measure the similarity, MI combined both spatial and structural information. By looking into the errors, it can be found that when Correlation was employed, some gestures were more easily confused with others, which led to *Substitution* error; when MI was employed, the partial part of some gestures was mistakenly recognized as an individual gesture, which led to *Insertion* error. Note that under the framework of Method I, the similarity measurement algorithm plays a key role in recognizing a gesture, i.e., the end time of a gesture was determined by the best matching score among all the motion segment candidates. Hence, the choice of Correlation or MI affected the final recognition results significantly. However, for Method II, DTW has been used as the first step to locate the endpoints of a gesture, and the similarity measurement methods were only used for fine tuning. Therefore, the errors were mainly caused by the DTW process for Method II, which was *Substitution* error in this case, no matter which similarity measurements were adopted.

With respect to computational cost, the time complexities of the three methods are $O(KMN)$, $O(K^2MN)$, and $O(KMN)$, for Methods I, II and III, respectively. Here, K is the number of gestures in the database, M is the length of the test gesture, and N is the maximum length of the Gesture Model in the database. Method III not only achieves the highest recognition rate, but also has the same lowest computational cost as that of Method I. Although Method II performs better than Method I, it is also more computationally expensive.

6.3. Hand gesture recognition

Besides the gesture type used in the previous two experiments, some tests were also conducted on hand gestures.

The most structured and widely studied hand gestures are the sign language used for communication by hearing impaired people. Analogous to phonemes being the basic units in speech, Stokoe defines a gesture in American Sign Language (ASL) by three basic units: location, hand shape, and movement [45]. A complete Stokoe's system includes 12 elemental locations, 19 hand shapes, and 24 movements (single and two-handed movements).

As a starting point, 12 hand shapes of the 19 elementary shapes were selected for recognition, irrespective to their location and movements. Here, the criterion used to choose the 12 hand shapes was that their contours must be distinctive from each other such that shapes could be differentiated from their contours. Fig. 13 shows the 12 hand shapes from nos. 1–12: A, B, B5, F, G, H, I, L, L3, V, W, and Y8. For simplicity, the gestures' locations and movements have not been taken into consideration. A huge amount of hand gestures could be derived from the combination of those hand shapes, given that a hand gesture is defined as a sequence of hand shapes. The task then was to recognize those elementary hand shapes.

Gesture Models were built based on the elementary hand shapes. To be specific, 10 sequences for each hand shape from a single subject were recorded. Gesture Models were then built for each of the 12 hand shapes using the same method as that of the above two experiments. The maximum scale s_{max} of the Gesture Model was set to 6, i.e., 30 frames.

For testing, the same person was asked to perform the hand shapes in arbitrary orders and 10 sequences were recorded. Each sequence spanned 600 frames, of which the number of hand shapes varied from 21 to 24. A total of 222 shapes were presented in those sequences, which were manually segmented and denoted as the ground truth. Fig. 14 shows two examples of such gesture sequences, which had 24 and 21 hand shapes, respectively. The numbers below each sequence correspond to the true hand shape numbers.

Method III, i.e., early-decision DP, has been applied to segment and recognize the hand gestures. Table 4 shows that the recognition rate for hand gestures achieved 82.0%. Although this result was not as high as that of recognizing continuous arm gestures, it is still very promising. In the hand gesture recognition task, only models for those predefined shapes were built; however, almost an unlimited number of non-gesture patterns, i.e., junk gestures,

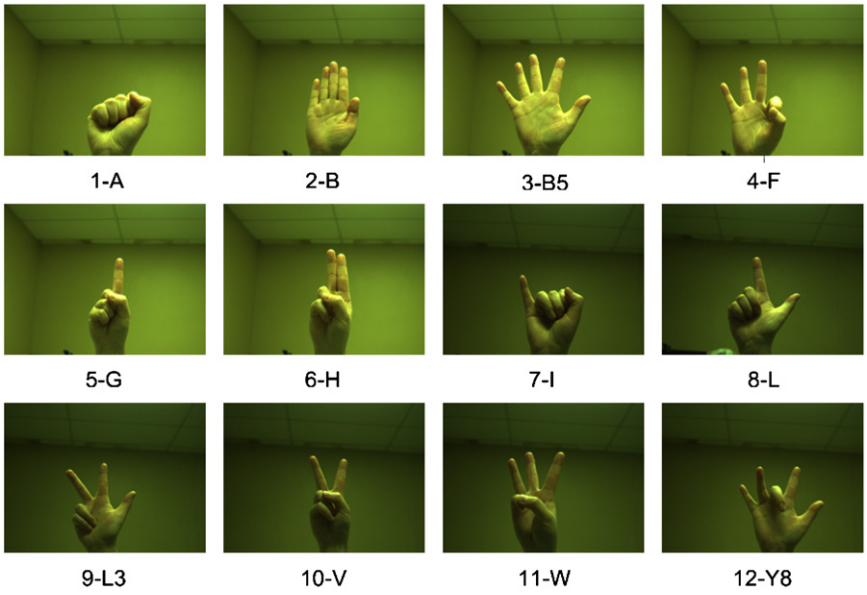

Fig. 13. 12 hand shapes from nos. 1–12.

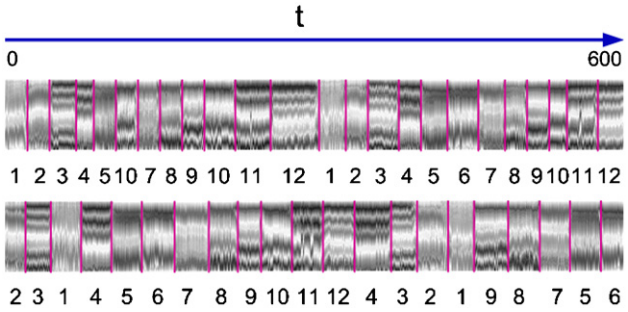

Fig. 14. Examples of hand gesture sequences: the first one contains 24 hand shapes, and the second one 21 hand shapes.

Table 4
Recognition rate for hand gesture test.

Total no.	Subs	Dels	Ins	Rec. rate (%)
222	17	18	5	82.0

Table 5
Comparison with other algorithms.

Methods	Alon et al. [4]	Kim et al. [17]	Lee et al. [5]	Our method
Gesture type	One hand	Upper body	One hand	Upper body
No. of gestures	10	8	10	8
User dependent	No	Yes	No	No
Handle junk gestures	No	No	Yes	No
Best rec. rate (%)	94.6	95.4	93.8	96.4

were present during the transition from one shape to another. Those junk gestures have been totally ignored in the experiments. Nevertheless, a recognition rate of over 80% was still achieved, which demonstrates the effectiveness of the proposed method. In order to further improve the performance, those junk gestures would have to be taken into account, as in [5], where a threshold model has been proposed to address this problem. We will discuss the junk gesture issue in the next section.

6.4. Comparison with state-of-the-art

There is unfortunately no common gesture database available, and so the majority of previously reported work in the literature create their own gesture databases for evaluation purposes. In addition, previously reported methods are not distributed for testing, and so it is difficult to compare the performance of our methods directly with that of others on common data. Table 5 lists some recent approaches which have similar testing databases to ours with respect to both the difficulty and the total number of the gestures tested. From Table 5 it can be seen that all methods achieved fairly high recognition rates, with ours the highest at 96.4%. Only the method of Lee et al. [5] specifically addressed the issue of junk gestures.

7. Discussion and conclusion

This paper has presented and compared three model-based methods to segment and recognize gestures from continuous video streams. Human non-rigid motion has been described by a Motion Signature, which uses contours changing over time to effectively capture the dynamic information of motion without exploring human body geometric details. A Gesture Model consists of a set of mean and variance images of a Motion Signature in a multi-scale manner, and not only is able to accommodate a wide range of spatio-temporal variation, but also has the advantage of requiring only a small amount of training data. All the three proposed methods, which either segment individual gestures by searching through all possible motion segments, or utilize a Dynamic Programming technique to detect the endpoints of a gesture, have achieved high recognition rates even when multiple gestures appeared in an arbitrary order.

Among the three approaches, Method III outperformed the other two by increasing the recognition rate by 7% compared to Method I, and by 5% compared to Method II. This demonstrates that the two-phase Dynamic Programming method is more effective in dealing with the endpoint localization issue than is the multi-scale matching and searching strategy used in the first two methods. The specific rules derived to determine when a gesture occurs work quite well. Those rules combine all the information available and are more reliable than methods depending solely on thresholds obtained from each individual gesture type [34].

An important issue in continuous gesture recognition is the presence of transition movements between adjacent real gestures, i.e., junk gestures, or *movement epenthesis* [30] in the field of continuous sign language recognition, which currently has not been taken into account in this work. The problem is prominent in the hand gesture experiment, where a large number of junk gestures exist when a hand transitions between shapes. The presence of those junk gestures considerably affects the accumulated distance values between the input signal and the gesture prototypes and brings more errors to the endpoint decisions. To improve segmentation accuracy, the issue of transition movements needs to be addressed. One way is to look for characteristic feature changes between meaningful gestures and meaningless non-gestures as in [15]. However, a more general algorithm is needed such as explicitly modeling movement epenthesis based on training data [19], or introducing an artificial threshold model from all trained Gesture Models [5].

Following the basic units definition for ASL by Stokeo [45], in addition to hand shape, location and movement of the hands have to be obtained from each frame and used as features. It has also been shown that language models, which provide very useful priors about gesture sequences, can improve the accuracy of continuous sign language recognition [36,37]. An interesting topic for future work will be exploring those directions to extend and apply the current system to the recognition of continuous ASL.

Acknowledgements

This work was supported by NSERC, and by the Government of Ontario, Canada, through the Premier's Research Excellence Award.

References

[1] S. Mitra, T. Acharya, Gesture recognition: a survey, IEEE Trans. Syst. Man Cybern Part C Appl. Rev. 37 (3) (2007) 311–324.
[2] S. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 873–891.
[3] V. Pavlovic, R. Sharma, T. Huang, Visual interpretation of hand gestures for human–computer interaction: a review, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 677–695.
[4] J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, A unified framework for gesture recognition and spatiotemporal gesture segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (9) (2009) 1685–1699.

[5] H.-K. Lee, J.H. Kim, An HMM-based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 21 (10) (1999) 961–973.
[6] B. Bauer, K.-F. Kraiss, Video-based sign recognition using self-organizing subunits, 16th International Conference on Pattern Recognition, vol. 2, 2002.
[7] A.F. Bobick, J.W. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 257–267.
[8] K. Derpanis, R. Wildes, J. Tsotsos, Hand gesture recognition within a linguistics-based framework, in: European Conference on Computer Vision, 2004, pp. 282–296.
[9] A. Mokhber, C. Achard, M. Milgram, Recognition of human behavior by space-time silhouette characterization, Pattern Recognition Lett. 29 (2008) 81–89.
[10] P. Peixoto, J. Goncalves, H. Araujo, Real-time gesture recognition system based on contour signatures, IEEE International Conference on Pattern Recognition, vol. 1, 2002, pp. 447–450.
[11] M.-H. Yang, N. Ahuja, M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1061–1074.
[12] J. Lichtenauer, E. Hendriks, M. Reinders, Sign language recognition by combining statistical DTW and independent classification, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 2040–2046.
[13] H. Yoon, J. Soh, Y.J. Bae, H.S. Yang, Hand gesture recognition using combined features of location, angle and velocity, Pattern Recognition 34 (2001) 1491–1501.
[14] Y. Zhu, G. Xu, D.J. Kriegman, A real-time approach to the spotting, representation, and recognition of hand gestures for human–computer interaction, Comput. Vision Image Understanding 85 (2002) 189–208.
[15] H. Kang, C.W. Lee, K. Jung, Recognition-based gesture spotting in video games, Pattern Recognition 25 (2004) 1701–1714.
[16] P. Morguet, M. Lang, Spotting dynamic hand gestures in video image sequences using hidden Markov models, International Conference on Image Processing, vol. 3, 1998, pp. 193–197.
[17] D. Kim, J. Song, D. Kim, Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms, Pattern Recognition 40 (2007) 3012–3026.
[18] M. Roh, B. Christmas, J. Kittler, S. Lee, Gesture spotting for low-resolution sports for video annotation, Pattern Recognition 41 (2008) 1124–1137.
[19] G. Fang, W. Gao, D. Zhao, Large-vocabulary continuous sign language recognition based on transition-movement models, IEEE Trans. Syst Man Cybern. Part A Syst. Humans 37 (1) (2007) 1–9.
[20] H. Li, M. Greenspan, Multi-scale gesture recognition from time-varying contours, International Conference on Computer Vision, vol. I, 2005, pp. 236–243.
[21] H. Li, M. Greenspan, Continuous time-varying gesture segmentation by dynamic time warping of compound gesture models, in: International Workshop on Human Activity Recognition and Modelling (HARAM), 2005, pp. 35–42.
[22] H. Li, M. Greenspan, Segmentation and recognition of continuous gestures, International Conference on Image Processing, vol. 1, 2007, pp. I365–I368.
[23] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[24] R.-H. Liang, M. Ouhyoung, A real-time continuous gesture recognition system for sign language, in: IEEE International Conference Automatic Face and Gesture Recognition, 1998, pp. 558–563.
[25] H. Silverman, D. Morgan, The application of dynamic programming to connected speech recognition, IEEE ASSP Mag. 7 (3) (1990) 6–25.
[26] C. Myers, L.R. Rabiner, A.E. Rosenberg, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, IEEE Trans. Acoust. Speech Signal Process. 28 (6) (1980) 623–635.
[27] A. Corradini, Dynamic time warping for off-line recognition of a small gesture vocabulary, in: IEEE ICCV Workshop on Recognition Analysis and Tracking of Faces and Gestures in Real-Time Systems, 2001, pp. 82–89.
[28] Y. Chen, Q. Wu, X.J. He, Using dynamic programming to match human behavior sequences, in: 10th International Conference on Control, Automation, Robotics and Vision, 2008, pp. 1498–1503.
[29] C. Myers, L.R. Rabiner, Connected digit recognition using a level-building DTW algorithm, IEEE Trans. Acoust. Speech Signal Process. 29 (3) (1981) 351–363.
[30] R. Yang, S. Sarkar, B. Loeding, Enhanced level building algorithm for the movement epenthesis problem in sign language recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[31] R. Yang, S. Sarkar, B. Loeding, Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming, IEEE Trans. Pattern Anal. Mach. Intell. 32 (3) (2010) 462–477.
[32] A.F. Bobick, A.D. Wilson, A state-based approach to the representation and recognition of gesture, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 1325–1337.
[33] J. Alon, V. Athitsos, S. Sclaroff, Accurate and efficient gesture spotting via pruning and subgesture reasoning, in: ICCV 2005 HCI Workshop, 2005, pp. 189–198.
[34] R. Oka, Spotting method for classification of real world data, Comput. J. 41 (8) (1998) 559–565.
[35] Y. Yacoob, M.J. Black, Parameterized modeling and recognition of activities, Comput. Vision Image Understanding 73 (2) (1999) 232–247.
[36] B. Bauer, H. Hienz, Relevant features for video-based continuous sign language recognition, in: IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 440–445.
[37] T. Starner, J. Weaver, A. Pentland, Real-time American sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1371–1375.
[38] S. Nayak, S. Sarkar, B. Loeding, Automated extraction of signs from continuous sign language sentences using iterated conditional modes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2583–2590.
[39] L.F. Costa, R.M. Cesar Jr., Shape Analysis and Classification: Theory and Practice, CRC Press, 2001.
[40] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (24) (2002) 509–522.
[41] L. Wang, W. Hu, T. Tan, Recent development in human motion analysis, Pattern Recognition 36 (2003) 581–601.
[42] S. Suzuki, K. Abe, Topological structural analysis of digitized binary images by border following, Comput. Vision Graphics Image Process. 30 (1985) 32–46.
[43] J. Pluim, J.B.A. Maintz, M.A. Viergever, Mutual-information-based registration of medical images: a survey, IEEE Trans. Med. Imaging 22 (8) (2003) 986–1004.
[44] P. Thénaz, M. Unser, Optimization of mutual information for multiresolution image registration, IEEE Trans. Image Process. 9 (12) (2000) 2083–2099.
[45] W.C. Stokoe, D. Casterline, C.C. Croneberg, A Dictionary of American Sign Language, Linstok Press, Washington, DC, 1965.

Hong Li received her Ph.D. from the Department of Electrical and Computer Engineering at Queen's University, Kingston, Canada in 2010. She earned her M.Eng. from National University of Singapore, and B.Eng. from Zhejiang University, China. Her research interests are in the area of computer vision, image processing and human motion recognition.

Michael Greenspan is a Professor and Head of the Department of Electrical and Computer Engineering at Queen's University, Kingston, Canada. Dr. Greenspan's research investigates problems of computer vision and robotics, with a focus on the development of efficient and robust object recognition and tracking methods.