J Multimodal User Interfaces
DOI 10.1007/s12193-014-0165-0

ORIGINAL PAPER

A novel set of features for continuous hand gesture recognition

M. K. Bhuyan · D. Ajay Kumar · Karl F. MacDorman · Yuji Iwahori

Received: 8 October 2012 / Accepted: 29 May 2014

Abstract Applications requiring the natural use of the human hand as a human–computer interface motivate research on continuous hand gesture recognition. Gesture recognition depends on gesture segmentation to locate the starting and end points of meaningful gestures while ignoring unintentional movements. Unfortunately, gesture segmentation remains a formidable challenge because of unconstrained spatiotemporal variations in gestures and the coarticulation and movement epenthesis of successive gestures. Furthermore, errors in hand image segmentation cause the estimated hand motion trajectory to deviate from the actual one. This research moves toward addressing these problems. Our approach entails using gesture spotting to distinguish meaningful gestures from unintentional movements. To avoid the effects of variations in a gesture's motion chain code (MCC), we propose instead to use a novel set of features: the (a) orientation and (b) length of an ellipse least-squares fitted to motion-trajectory points and (c) the position of the hand. The features are designed to support classification using conditional random fields. To evaluate the performance of the system, 10 participants signed 10 gestures several times each, providing a total of 75 instances per gesture. To train the system, 50 instances of each gesture served as training data and 25 as testing data. For isolated gestures, the recognition rate using the MCC as a feature vector was only 69.6 % but rose to 96.0 % using the proposed features, a 26.1 % improvement. For continuous gestures, the recognition rate for the proposed features was 88.9 %. These results show the efficacy of the proposed method.

Keywords Human–computer interaction (HCI) · Gesture recognition · Motion chain code (MCC) · Conditional random fields (CRF)

M. K. Bhuyan (✉) · K. F. MacDorman
School of Informatics and Computing, Indiana University Purdue University (IUPUI), 535 West Michigan St., Indianapolis, IN 46202, USA
e-mail: mkbhuyan@iupui.edu

K. F. MacDorman
e-mail: kmacdorm@indiana.edu

D. Ajay Kumar
Department of Electronics and Electrical Engineering, IIT, Guwahati 781039, India
e-mail: ajay.k@iitg.ernet.in

Y. Iwahori
Department of Computer Science, Chubu University, 1200 Matsumoto-cho, Kasugai 487-8501, Japan
e-mail: iwahori@cs.chubu.ac.jp

1 Introduction

Gestures provide an attractive, user-friendly alternative to using an interface device like a keyboard, mouse, and joystick in human–computer interaction (HCI). Accordingly, the basic aim of gesture recognition research is to build a system that can identify and interpret specific human gestures automatically and employ them to convey information (i.e., for communicative use as in sign-language) or to control devices (i.e., manipulative use as in controlling robots without any physical contact).

One of the most important requirements for sign language recognition is that natural gesturing be supported by the recognition engine so that a user can interact with the system without any restrictions. Since a sequence of gestures is generally mixed with coarticulation and unintentional movements, these non-gestural movements should be eliminated from an input video before the identification of each gesture in the sequence. Movement epenthesis is a non-gestural movement between gestures, and gesture coarticulation is

Published online: 02 July 2014

the modification of the beginning or end of a sign because of the sign that preceded or suceeded it, respectively [1,2]. In more natural settings, the gestures of interest are embedded in a continuous stream of motion, and their occurrence must be detected. This is precisely the goal of gesture spotting, namely, to locate the starting point and the endpoint of a gesture pattern and to classify the gesture as belonging to one of the predetermined gesture classes. Once the gesture boundaries are known, a gesture in the sequence can be conveniently extracted. However, the process of gesture boundary detection is not trivial for the following reasons:

- Gesture boundaries vary from one signer to another.
- Gesture boundaries are sequence dependent; in particular, gesture segmentation is heavily influenced by higher-level cognitive processes and by the context of each gesture.
- It is impossible to enumerate all gestures. Traditional computer vision approaches characterize a gesture as a series of poses. Clearly, given the virtually infinite number of poses that the human body can assume, a generic model-based approach to gesture segmentation is not viable.

Generally, a gesture starts and ends with the hand at a standstill. That is, a signer generally starts making a sign from a *pause* state and ends in a *pause* state even when gesturing continuously [3]. Based on this observation, we propose to use the hand motion information to locate the boundary points of each gesture in a continuous stream of gestures. A boundary point is detected whenever the hand pauses during gesturing. For gestures having global motion only or gestures having both global and local motions, the gesturing hand traverses through space to form a gesture trajectory. After a gesture trajectory is completed, the hand pauses for a while and then moves with very high velocity to the starting position of the next trajectory. Based on this, we propose to detect movement epenthesis—and distinguish it from a gesture stroke—by observing the motion of the hand between *pauses* in the input hand motion video. Whereas a movement epenthesis phase is simply a fast hand stroke, a gesture phase can be divided into three motion stages: preparation, stroke, and retraction. However, the scheme will result in the incorrect spotting of gestures under the conditions listed below:

- It is not correct to assume that between gestures there is always some non-gestural hand movement. For example, there is generally no extra movement if a gesture ends in the same position or pose at which the next gesture begins. In this case, the two gestures are adjoined to each other in the sequence with a common *pause* indicating the end of the first gesture and the start of the next gesture.
- For a sequence of static hand poses (fluent finger spelling), there is generally no motion during the gesturing period while the hand may move in between two gesture poses because of movement epenthesis. That means a *pause* itself in the sequence corresponds to a gesture sign as if the starting point and endpoint of the gesture have merged together.
- For gestures involving global hand motion, there may be some *pauses* within a single gesture. When the hand traverses through space, it makes one or more hand strokes to build up a complete gesture trajectory with *pauses* between the strokes.

The proposed method is described in more details in the following sections.

2 The state of the art

A threshold model based on a hidden Markov model (HMM) was proposed to recognize and spot the gestures [4] in which motion chain code (MCC) is used as the feature. The system was trained with a huge number of isolated samples. The gesture is spotted when it crosses the adaptive threshold and the boundary points are detected using the Viterbi back-tracking algorithm. The major limitation of this method is that as the gesture vocabulary increases the number of states increases. Most errors come from the failure of hand extraction that distorts the hand trajectory and also the MCC. Our proposed method focuses on solving this problem by replacing the MCC with other trajectory-based features.

A gesture spotting technique was proposed using a finite state machine (FSM) [5]. The features used are various motion parameters like the velocity of the hand. The velocity of the hand is less at the beginning and end of the gesture, and the hand abruptly moves during movement epenthesis. However, this method cannot be used as a generalized model for all the gestures.

Gesture spotting with a threshold model based on conditional random fields (CRF) combines motion-based and location-based features [6]. The MCC of the hand trajectory is used as a motion-based feature. As with other methods, variations in the extracted MCC greatly affect the performance of the recognizer.

A framework that simultaneously spots and recognizes gestures by using the density functions of the states for each of the gesture classes was proposed in [7]. A 4D position-flow vector is used as the feature, which contains the pixel position of the centroid of the hand and its two dimensional velocity. This method is also not applicable to all kinds of gestures.

A sign language recognition system was developed using three appearance-based features: PCA, kurtosis position, and the MCC [8]. The hand is tracked and the trajectory is represented by MCC. These three features are given to three separate HMM networks for recognition. Thus, the overall recognition accuracy also depends on MCC.

Though the *segmentation* and *coarticulation detection* are the main open research issues for continuous hand gesture recognition, few vision-based approaches have been reported prior to this work. The techniques developed so far for coarticulation detection have not always been successful for a wide range of gesture vocabulary. Although powerful classifiers for labeling the sequential data exist, feature selection is key to obtaining a high recognition rate. In gesture recognition, the features must be selected so as to enable the classifier to unambiguously recognize and segment the continuous gestures. As explained earlier, most of the existing approaches use the motion chain code as a feature. The MCC is a rotation invariant feature for object recognition. But in gesture recognition, many variations occur in the extracted MCC even when the same gesture is performed repeatedly. This is because of ambiguity in hand image segmentation and trembling of the gesturing hand.

As an example, Fig. 1 shows frames of the segmented hand regions of the gesture trajectory *Eight*. The area and the orientation of the segmented hand differs from frame to frame. This causes the extracted trajectory to deviate from the actual trajectory, resulting in drastic variations in the MCC, which was encoded from the information of the motion trajectory. This variation in the MCC causes difficulty in modeling a gesture. Figure 2 shows the extracted gesture trajectory varies from the actual one. Hence, for a particular gesture, the MCC will not be unique. To overcome this problem, we propose a much smoother chain code feature than the conventional MCC. The new feature gives better results. Additionally, two other features are proposed to cope with the problem of movement epenthesis.

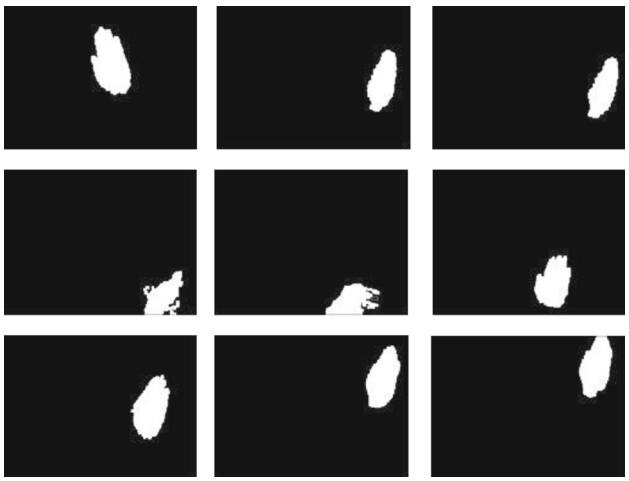

Fig. 1 Various frames of the segmented hand of the gesture *Eight*

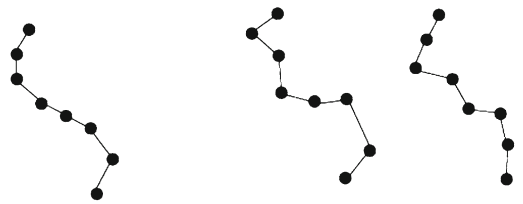

Fig. 2 Examples of trajectories of isolated gestures

3 Proposed system

This paper focuses on the recognition of continuous digits gesticulated by the user. These gestures, shown in Fig. 3, are dynamic, consisting only of global hand motion. These gestures are performed continuously, so movement epenthesis and coarticulation come into consideration as the gesturer has to move his hand from the endpoint of one gesture to the starting point of other gesture. Thus, the recognition system should be capable of detecting the meaningful gestures from the continuous stream of gesticulation. Due to coarticulation, the appearance of a sign/gesture, especially at the beginning and end, can be significantly different under different sentence contexts, which makes the recognition of gestures in sentences hard. In other words, when gestures are produced continuously, each gesture may be affected by the preceding gesture, and sometimes by the gesture that follows it. However, movement epenthesis occurs when the hand moves very fast from the end position in a gesture to the starting position in the following gesture. This implies that these movements are deliberate and are essential for connecting two gestures in a sequence. A pictorial representation of movement epenthesis and the embedded gestures is shown in Fig. 4.

In our method, we use CRF to classify the sequential data (feature vector), because they can model the gestures by considering the dependencies in the sequential data. CRFs have the ability to relax the independence assumption of hidden Markov models [9,10]. However, an HMM is a stochastic automaton that models the sequential data by generating the observation while transiting from one state to another state [11]. Whereas, CRF is a non-generative finite state model with a single exponential for joint probability of the label sequence given an observation sequence. Most of the existing gesture spotting approaches in the literature rely on dynamic time warping (DTW), neural networks (NN), or HMM.

Once a particular gesture has been recognized, it needs to be mapped to a corresponding action, for example, to control a robot or activate a window menu. A gesture-based HCI system also allows a person to communicate with a computer via

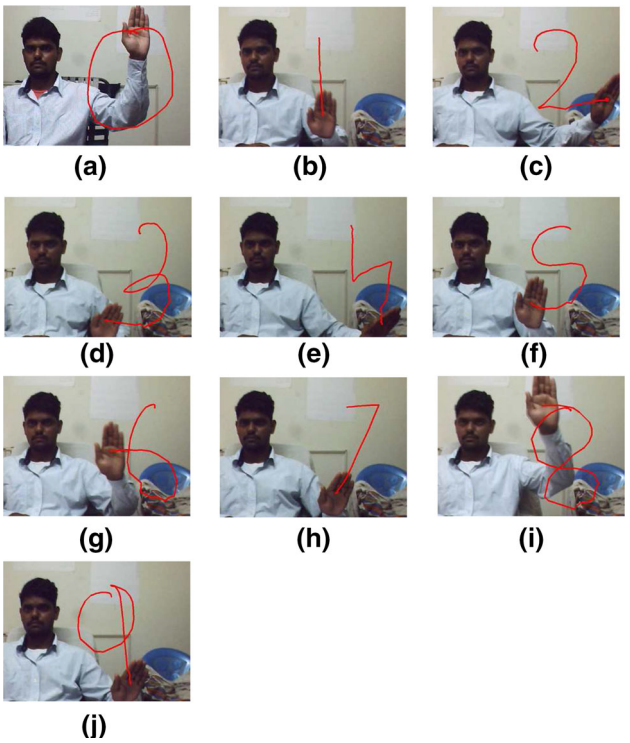

Fig. 3 Examples of trajectories of isolated gestures

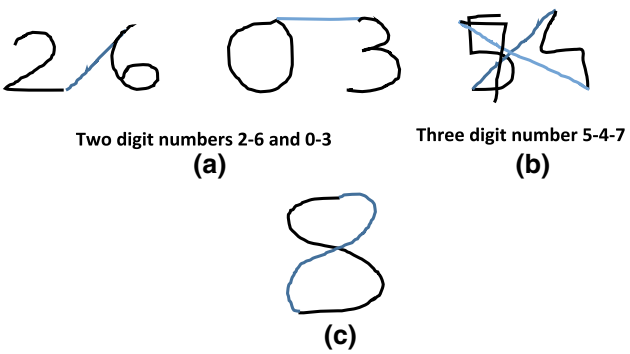

Fig. 4 Problems in geture recognition include movement epenthesis, indicated by *dotted lines* in (**a**) and (**b**), and the embedding of one gesture in another. In **c** *Five* is embedded in *Eight*

sign language and, thus, enables those with hearing impairment to interact with a computer more easily. Our method is basically intended for these kinds of HCI applications. The proposed scheme for continuous hand gesture segmentation and subsequent recognition is described in the sections to follow.

3.1 Hand segmentation and gesture trajectory estimation

Hand segmentation is typically a first step in tracking the movement of the hand. In our method, the face region in the frame is removed by using a face detection algorithm. Skin color-based segmentation is used to extract the skin regions

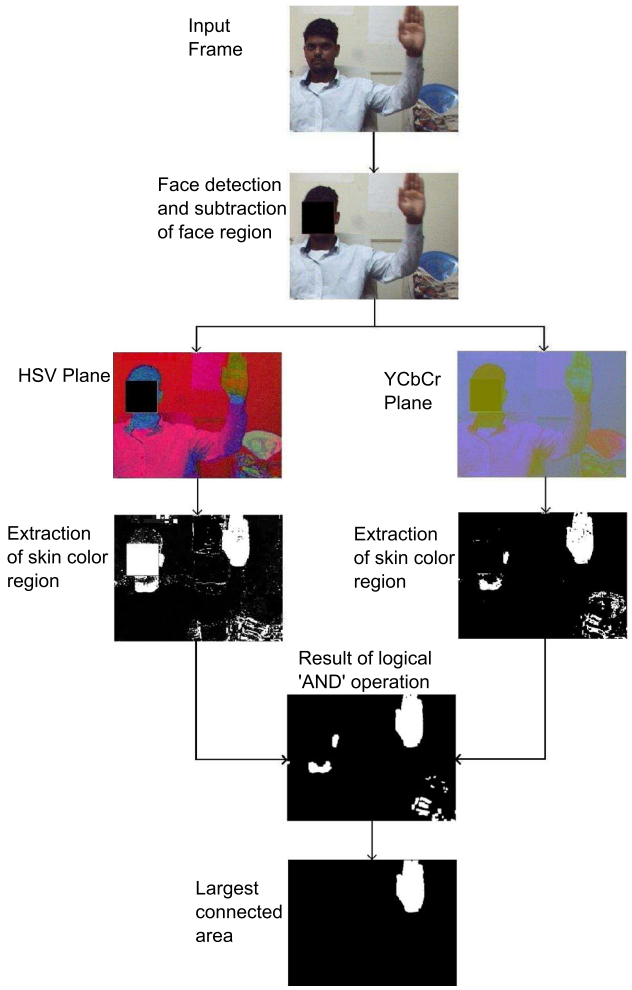

Fig. 5 Hand segmentation by the proposed method

of the hand from the video frame. Skin color is clustered in a very small region in a color space so that skin regions can easily be separated by selecting a proper threshold. In addition, color spaces are selected that are less susceptible to changes in lightning. HSV and YCbCr are two color spaces that separate the chrominance ([H S] or [Cb Cr]) and luminance (V or Y) components [12]. So, after removing the face region, the RGB image is converted to HSV and YCbCr images, and subsequently a threshold is applied to the chrominance components of both the HSV and YCbCr color spaces. Finally, a logical *AND* operation is performed between them to obtain the likely skin region.

After getting the skin color segmented regions, the largest connected segment corresponds to the palm region of the hand. Figure 5 shows various intermediate results. After determining the binary alpha plane corresponding to the palm region of the hand, moments are used to find the center of the hand. The 0th and the 1st moments are defined as

$$M_{00} = \sum_x \sum_y I(x, y), \; M_{10} = \sum_x \sum_y x I(x, y),$$

$$M_{01} = \sum_{x}\sum_{y} yI(x, y) \tag{1}$$

Subsequently, the centroid is calculated as

$$x_c = \frac{M_{10}}{M_{00}} \quad \text{and} \quad y_c = \frac{M_{01}}{M_{00}} \tag{2}$$

In the above equations, $I(x, y)$ is the pixel value at the position (x, y) in the image. Since the background pixels are assigned **0**, the centroid of the hand in a frame is also the centroid of the total frame. Therefore, in moment calculations, we may either take the summation over all pixels in the frame or over only the hand pixels.

Finally, the gesture trajectory is formed by joining all the calculated centroids in a sequential manner. However, this trajectory may be noisy for the following reasons:

- Points too close
- Isolated points far from the correct trajectory due to a change in the hand shape
- Unclosed endpoints
- Hand trembling
- Unintentional movements

Therefore, to reduce these effects, the final trajectory is smoothed out by considering the mean value of a specified point and its two neighboring points, i.e.,

$$(\hat{x_t}, \hat{y_t}) = ((x_{t-1} + x_t + x_{t+1})/3, (y_{t-1} + y_t + y_{t+1})/3) \tag{3}$$

So, a dynamic hand gesture (DG) can be interpreted as a set of points in a spatiotemporal space:

$$DG = \{(\hat{x_1}, \hat{y_1}), (\hat{x_2}, \hat{y_2}), \ldots\ldots, (\hat{x_t}, \hat{y_t})\} \tag{4}$$

3.2 Extraction of proposed trajectory-based features

Given the sequence of centroid of segmented hand, three features are extracted as shown in Table 1. The proposed three dimensional feature vector is extracted to use the CRF effectively, so that it can spot the gestures and recognize them.

As shown in Fig. 6, feature C_E is extracted from the angle of the major axis of the ellipse which is fitted to the previous r points $P = \{X_t\}_{t=T-r}^{T}$ of the motion trajectory, where $X_t = (x_t, y_t)$. The motivation behind this idea is to obtain a chain code that preserves the shape of the motion trajectory even if the extracted trajectory is not smooth. The ellipse is fitted using least squares fitting of conic sections. Let us assume an ellipse with a set of parameters $A = [a\, b\, c\, d\, e\, f]^T$ as

$$E(A, X) = ax^2 + bxy + cy^2 + dx + ey + f$$

Table 1 Proposed features

Features	
C_E	Chain code obtained by the angle of orientation of the ellipse
L_E	Length of the major axis of ellipse (small/medium/large)
P_H	Position of the hand (top/middle/bottom)

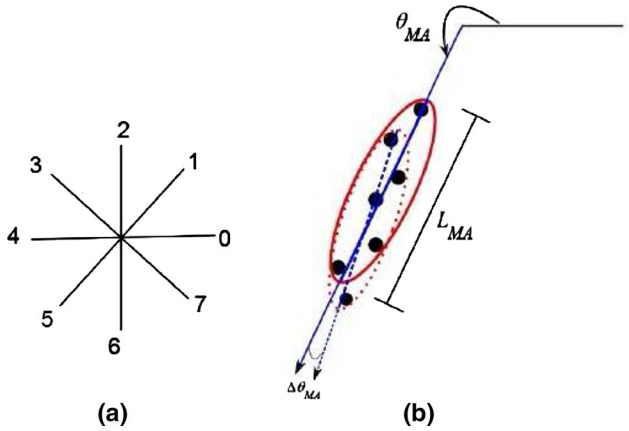

Fig. 6 Extraction of features C_E and L_E

The algebraic distance $E(A, X)$ is the product of $\chi_t = [x_t^2\, x_t y_t\, y_t^2\, x_t\, y_t\, 1]$ and A.

$$E(A, X) = [x_t^2\, x_t y_t\, y_t^2\, x_t\, y_t\, 1][a\, b\, c\, d\, e\, f]^T = \chi_t A$$

The parameter set A is found using the least-squares technique that minimizes a distance metric $\epsilon(A)$ between the data points and the ellipse E.

$$\epsilon(A) = \sum_{t=T-r}^{T} E(A, X_t)^2 = \|DA\|^2$$

where, $D = \left[\chi_{T-r}\ \chi_{T-r+1} \ldots \chi_T\right]$ is the design matrix. To avoid the trivial solution $A = 0_6$, different constraints are applied to the parameter A. We chose the computationally efficient algorithm given in [13] for ellipse fitting. The minimization of $\|DA\|^2$ subject to the constraint $4ac - b^2 = 1$ gives only one solution, which is an ellipse. The counter-clockwise angle of the major axis of that ellipse (θ_{MA}) is used to find our proposed chain code/feature C_E, whereas a conventional MCC is calculated with the slope of the lines joining to successive points.

In our proposed method, starting from the first trajectory point (at the beginning of a gesture), an ellipse is fitted to the set of points of the motion trajectory until r points are enclosed. Then, the straight line joining the two extreme points of the major axis of the ellipse is determined. Next, as shown in Fig. 6, the said ellipse is fitted to the next r points starting from the second trajectory point, and the straight line joining the two extreme points of the major axis of the ellipse is determined. Finally, the angle $(\Delta\theta_{MA})$ is calculated from

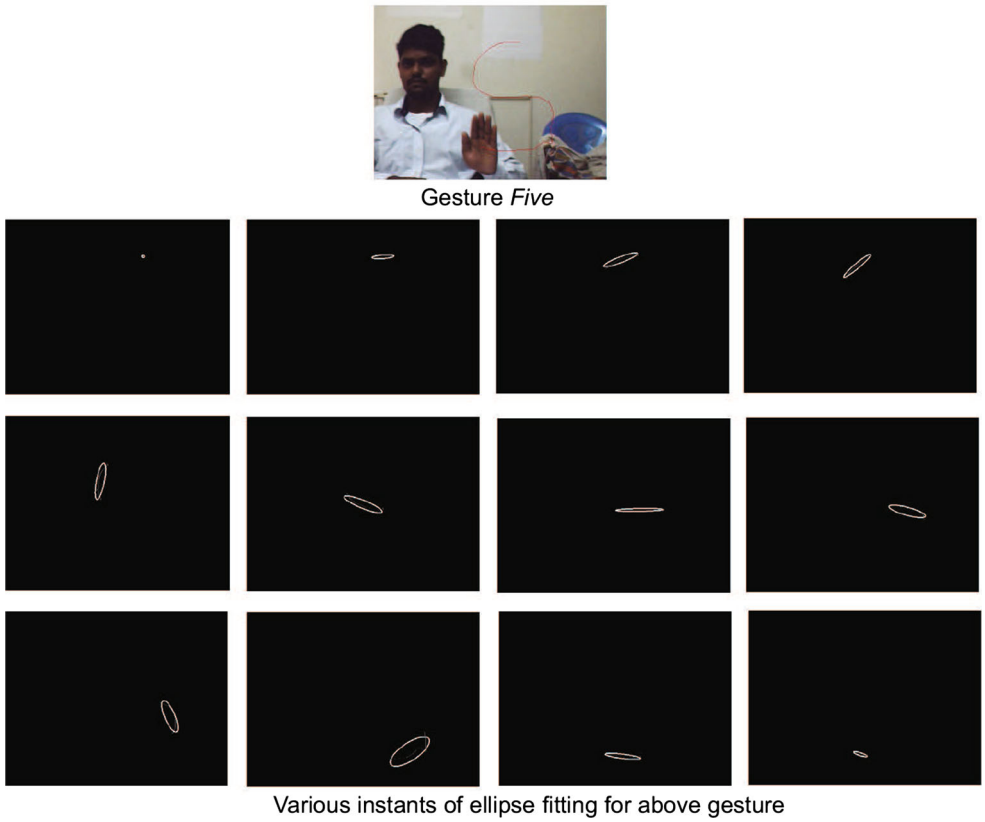

Fig. 7 Ellipse fitting to gesture trajectory *Five* for $r = 6$

these two lines, which is subsequently used for determining the proposed chain code/feature. This process is performed for all the points of the motion trajectory. In our experiment, the value of r is set to 6. It is experimentally observed that the hand stops for at least six frames at gesture boundaries, and the length of the major ellipse is very small for the endpoints. This consideration additionally helps us to detect the endpoints (gesture spotting). Figure 7 shows various instants of ellipse fitting corresponding to gesture trajectory five.

As explained in Sect. 1, the conventional MCC fails to describe the original track of the gesture due to segmentation ambiguities. But the new feature C_E preserves the shape of the gesture trajectory, because in this case the rate of change in the orientation angle ($\Delta\theta_{MA}$) is much lower. Hence, it results in a smoother chain code. Table 2 shows the difference between MCC and C_E for the gesture *One*. The same gesture is performed three times and the parameters MCC and C_E are extracted. Clearly, the proposed feature C_E nicely encodes the shape of the gesture trajectory.

Next, the feature L_E is obtained from the length of the major axis of the ellipse L_{MA}. As shown in Fig. 7, L_{MA} decreases at the endpoints of a gesture. It is a ternary-valued feature [(S) or (M) or (L)], which is obtained by setting the thresholds T_a and T_b. It indicates whether the length of L_{MA} is small (S) or medium (M) or large (L). Hence, the temporal information of the hand trajectory can be known and the

Table 2 Example showing the difference between MCC and C_E for gesture trajectory *One*

MCC	C_E
676556566566066066666667 6055	666666666666666666666666666
3567007676660666665667620	344666666666666666666666
5665666766660666666672	566666666666666666666

information helps in gesture spotting. This also helps in internal modeling of gestures because few gestures have intermediate stoppages in between the starting and the endpoints of the gestures, namely, motion trajectories of the gestures *three*, *two*, *four*, *seven*, and *nine*. The feature L_E is extracted as follows:

$$L_E = \begin{cases} S, & L_{MA} \leq T_a \\ M, & T_a < L_{MA} \leq T_b \\ L, & L_{MA} > T_b \end{cases}$$

Finally, the feature P_H represents the position of the hand in the frame. It is also a ternary-valued feature (T or M or B). As shown in Fig. 8, each image frame is horizontally divided into three regions, which are called as the top, middle, and bottom. Subsequently, P_H in each frame is calculated by setting thresholds T_c and T_d for the x coordinate of the centroid of the hand. This feature is targeted to detect

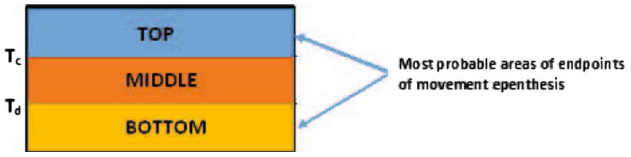

Fig. 8 Extraction of feature P_H

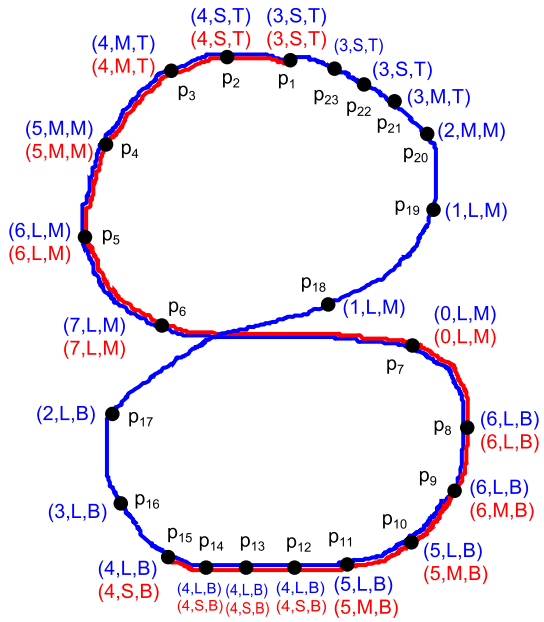

Fig. 9 An example showing the proposed set of features (C_E, L_E, P_H) for the gestures *Five*, indicated by the *red-colored* trajectory, and *Eight*, indicated by the *blue-colored* trajectory

the movement epenthesis. The boundaries of the movement epenthesis stroke are generally either in the top or bottom position of a video frame, whereas the middle region always contains the body of the gesture. The feature P_H is derived as follows:

$$P_H = \begin{cases} T, & x_t \leq T_c \\ M, & T_c < x_t \leq T_d \\ B, & x_t > T_d \end{cases}$$

The proposed features for the gestures 'Eight' and 'Five' are shown in Fig. 9. It is evident that these three features may be the same for both of the gestures until point p_8, but the feature L_E changes afterwards. Hence, the proposed features can detect a gesture when it is embedded in another gesture.

3.3 Proposed gesture recognition scheme

Finally, as shown in Fig. 10, the extracted features C_E, L_E, and P_H are applied to the proposed classifier. In our experiment the train/test sequence $o = \{o_1, o_3, o_3, \ldots, o_m\}$ is associated with the label sequence $w = \{w_1, w_3, w_3, \ldots, w_m\}$. The probability of the label sequence w given o is calculated by using a log-linear exponential model:

$$p(w|o) = \frac{1}{z(o)} \exp\left(\sum_{j=1}^{p} \alpha_j t_j\,(w_{i-1}, w_i, o, i) + \sum_{k=1}^{q} \beta_k s_k\,(w_i, o, i)\right)$$

$$z(o) = \sum_{w} \exp\left(\sum_{j=1}^{p} \alpha_j t_j\,(w_{i-1}, w_i, o, i) + \sum_{k=1}^{q} \beta_k s_k\,(w_i, o, i)\right)$$

where, $t_j\,(w_{i-1}, w_i, o, i)$ is a transition feature function of the observation sequence and the labels at positions i and $i-1$, $s_k\,(w_i, o, i)$ is a state feature function of the label at position i and the observation sequence, p and q are the total number of features for transition feature functions and state feature functions, α_j and β_k are weights to the exponent which are estimated from training data by using the maximum entropy criterion, and $z(o)$ is a normalization factor to ensure that $\sum_w p(w|o) = 1$.

The transition and state feature functions are calculated using the binary-valued predicates on the observation data. An example of such a predicate in our experiment is given by

$$g\,(o, i) = \begin{cases} 1 & \text{if } o \text{ is at the position } i \text{ defined by the} \\ & \text{feature value } (4,\ S,\ B) \\ 0 & \text{otherwise.} \end{cases}$$

An example of the state and the transition feature functions that takes on the value of the predicates are given by

$$t_j\,(w_{i-1}, w_i, o, i) = \begin{cases} g\,(o, i) & \text{if } w_{i-1} = \text{two} \\ & \text{and } w_i = \text{one} \\ 0 & \text{otherwise} \end{cases}$$

$$s_k\,(w_i, o, i) = \begin{cases} g\,(o, i) & \text{if } w_i = \text{one} \\ 0 & \text{otherwise} \end{cases}$$

These transition and state feature functions help the CRF to model the data with arbitrary dependencies. An iterative algorithm called improved iterative scaling (IIS) is then used to estimate the parameters $\theta = \{\alpha_1, \alpha_2, \ldots; \beta_1, \beta_2, \ldots\}$ from the training data $\Omega = \left\{\left(o^t, w^t\right)\right\}_{t=1}^{T}$ that maximizes the log-likelihood objective function, which is found as

$$L\,(\theta) = \sum_{t=1}^{T} \log p_\theta\left(w^t|o^t\right) \propto \sum_{o,w} p\,(o, w) \log p_\theta\,(w|o)$$

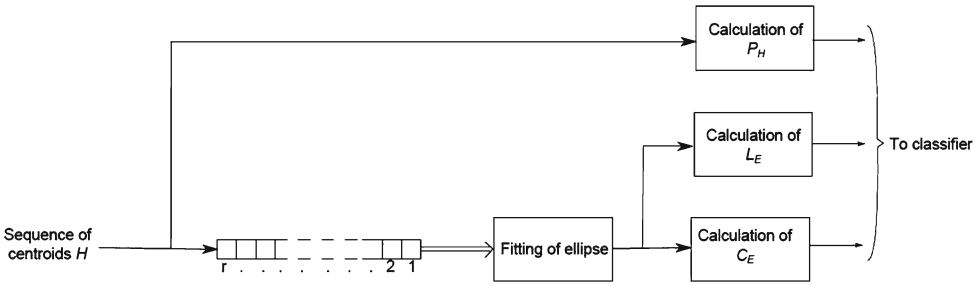

Fig. 10 Block diagram showing proposed set of features C_E, L_E and P_H

The conditional probability of the label sequence is computed efficiently by using the matrices. Let there be $(n+1)$ labels in the label sequence with a starting and stopping label w_0 = **start** and w_{n+1} = **stop**. Then $p_\theta\,(w|o)$ is calculated using a set of $n+1$ matrices. In this, $\{M_i\,(o)\,|i=1,2,\ldots,n+1\}$ where, $M_i\,(o)$ is a $|y \times y|$ matrix with elements of the form

$$M_i\left(w', w|o\right) = \exp\left(\sum_j \alpha_j t_j\,(w_{i-1}, w_i, o, i) + \sum_k \beta_k s_k\,(w_i, o, i)\right)$$

The normalization factor $z_\theta(o)$ is the product of these matrices

$$z_\theta(o) = (M_1\,(o)\,M_2\,(o)\,\ldots\ldots\,M_{n+1}\,(o))_{\text{start, stop}}$$

And the conditional probability of the label sequence w is given as

$$p_\theta\,(w|o) = \frac{\prod_{i=1}^{n+1} M_i\,(w_{i-1}, w_i|o)}{\left(\prod_{i=1}^{n+1} M_i\,(o)\right)_{\text{start, stop}}}$$

In our proposed method, CRF is trained with one label for each gesture and an extra label is added to determine the non-gestural movements. Apart from the isolated gestures, all possible non-gestural movements are trained. To model the dependencies of non-gestural movements with gestures, a combination of gesture and non-gestural observation sequences are used in the training data.

4 System performance evaluation

The proposed system was tested in real-time on an Intel® Core I3-based personal computer. The input images are captured by a CCD camera at a resolution of 480 × 480 pixels. The system is evaluated using both isolated and continuous gestures. During the evaluation, we use a comparatively simple background as shown in Fig. 3. The background enhances hand segmentation performance. Moreover, all the experiments are performed with nearly constant uniform illumination, and all the gestures under consideration are performed with one hand only.

In the evaluation, we used gesture sequences corresponding to the numerals 0 to 9, as shown in Fig. 3. To obtain the feature L_E, the thresholds T_a and T_b were set to 60 and 180, respectively. To extract the feature P_H, the captured image frame is divided into three equally spaced horizontal blocks. As the frame resolution is 480 × 480, the thresholds T_c and T_d were set to 160 and 320, respectively. To ensure spatiotemporal variability, each gesture was performed by 10 signers. Even if the same person tries to perform the same sign twice, small variation occur in the speed and position of the hands. In total, 50 isolated samples per gesture were used to train the CRF. Subsequently, the same CRF was again trained with another 50 samples per gesture class with each feature sequence corresponding to the selected gesture appended with the possible movement epenthesis segments. This training method creates an extra label in the CRF to handle movement epenthesis as the CRF learns the dependencies of movement epenthesis for the gestures. In addition, by setting the window size to 3, the CRF is designed to use only three past and future observations to predict the current state. Motion trajectories extracted by our method are shown in Fig. 11.

4.1 Isolated gesture recognition

The classification results corresponding to different gesture sequences are shown in Table 3. The overall recognition rate for our proposed method is 96.0 % as compared with 69.6 % when the MCC is used as a feature vector, a 26.1 % increase. The improvement is mainly because of the smoother chain code obtained by our proposed feature C_H. However, a relatively low recognition rate is reported for the gesture *One*, which might be due to the inability of CRFs to model the long-range dependencies. This can be overcome by increasing the window size, but this generates many features, which increases the computational complexity. Also, we note that our recognizer misclassifies the gestures *One* and *Seven* in some critical conditions. The misclassification occurs when the hand stops too long in the middle of the gesture *Seven*.

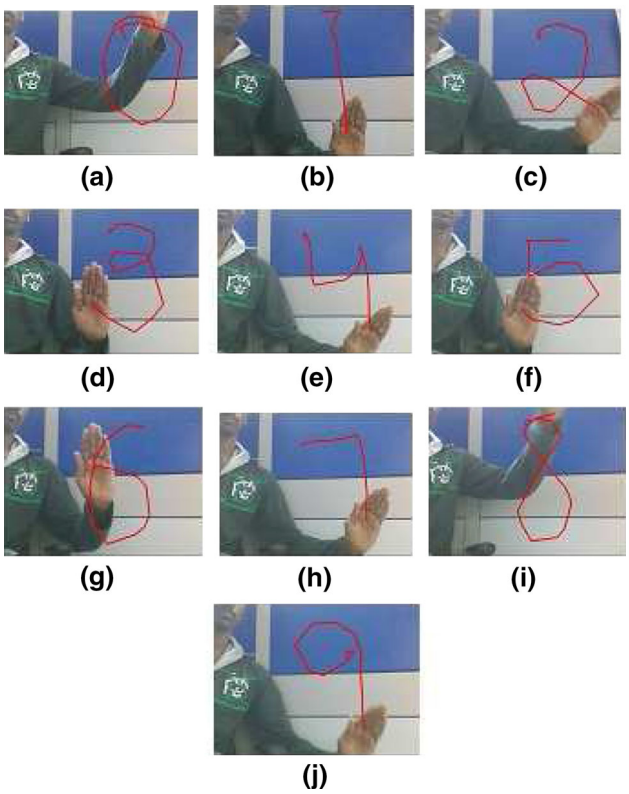

Fig. 11 Motion trajectories extracted by the proposed method

4.2 Continuous gesture recognition

For continuous gesture recognition, 40 video clips of different continuous gestures that contains movement epenthesis are considered and 216 sample gestures are used to test the system. We have only considered a few typical possible movement epenthesis and the embedded gesture cases for the evaluation of the proposed algorithm. Some of the test gesture trajectories are shown in Fig. 12. Three parameters are used to evaluate the performance of the proposed continuous gesture recognizer. They are labeled as insertion error, deletion error, and substitution error. The insertion error occurs when the spotter reports a nonexistent gesture. The deletion error occurs when the spotter fails to detect a gesture. The substitution error occurs when the spotter falsely classifies a gesture. The detection ratio is the ratio of correctly recognized gestures over the number of input gestures. Table 4 shows the performance of our proposed system in terms of the above mentioned parameters. The proposed system recognizes the continuous gestures with an accuracy of 88.9 %. As shown in Fig. 12, the recognized gestures are displayed in the form of blue-colored text appearing in the top right corner of the image, where *?* indicates the movement epenthesis present in between the gestures. Additionally, as shown in Fig. 13, a gesture-based numeric calculator is implemented to highlight a possible application of our proposed method.

5 Conclusions

One critical issue in continuous gesture recognition research is to identify meaningful gestures in a continuous stream of body movements. This may be accomplished by spotting precisely when a gesture in the sequence starts and ends. This is the goal of gesture spotting. Gesture spotting is essential for a recognition system to work continuously without need of human intervention. In particular, only if gesture spotting is supported in a vision-based interface, it is possible for a user to interact with the recognition platform using natural gestures without any restrictions. By spotting gestures in a

Table 3 Experimental results: isolated gesture recognition rates

Class label	No. of training samples per class: 50 No. of test samples per class: 25 No. of test pattern assigned to predefined class										Acc. rate %
	Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	
Zero	25	0	0	0	0	0	0	0	0	0	100
One	0	21	0	0	0	0	0	4	0	0	84
Two	0	0	25	0	0	0	0	0	0	0	100
Three	0	0	0	25	0	0	0	0	0	0	100
Four	0	0	0	0	23	2	0	0	0	0	92
Five	0	0	0	0	0	25	0	0	0	0	100
Six	0	0	0	0	0	0	25	0	0	0	100
Seven	0	4	0	0	0	0	0	21	0	0	84
Eight	0	0	0	0	0	0	0	0	25	0	100
Nine	0	0	0	0	0	0	0	0	0	25	100
Avg.											96.0

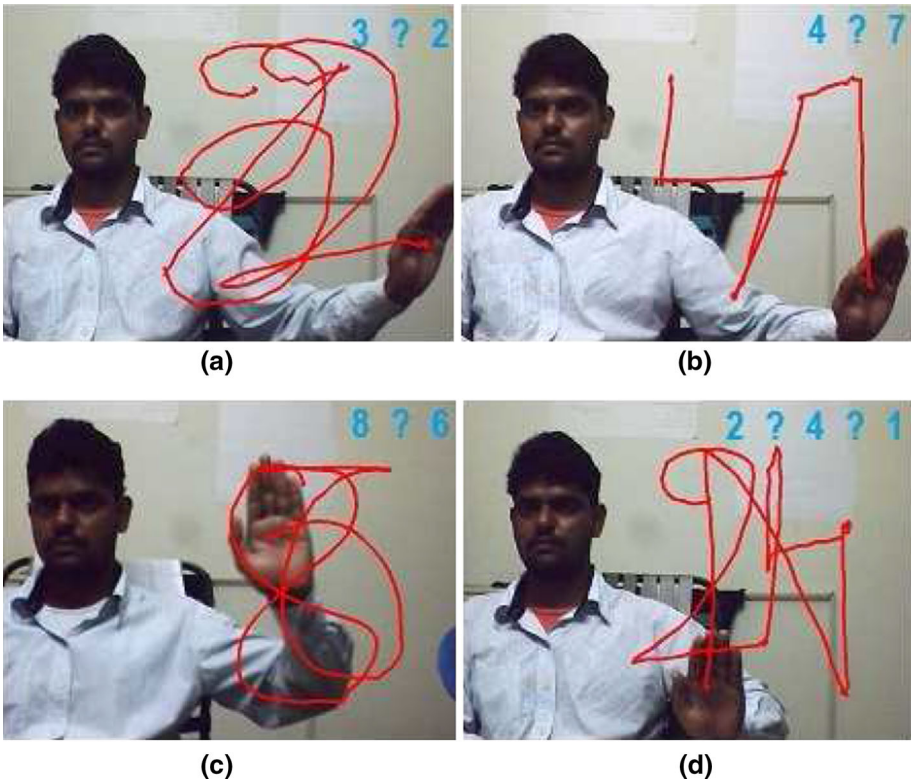

Fig. 12 Continuous gestures: **a** Three-Two, **b** Four-Seven, **c** Eight-Six, **d** Two-Four-One

Table 4 Continuous gesture recognition using the proposed features

Gesture	No. of gestures	Insert	Delete	Substitute	Correct	Detection (%)
Zero	16	0	2	3	11	68.75
One	20	0	0	4	16	80.00
Two	25	0	0	0	25	100.00
Three	26	0	0	0	26	100.00
Four	21	0	0	0	21	100.00
Five	19	1	0	1	18	94.74
Six	20	0	3	5	12	60.00
Seven	25	1	1	3	21	84.00
Eight	29	0	0	1	28	96.55
Nine	25	2	1	1	23	92.00
Total	226	4	7	18	201	88.94

continuous video stream, we can automatically remove unintentional movements arising between gestures.

Again, coarticulation detection is one of the main challenges in continuous gesture recognition. It is difficult to distinguish between a valid gesture phase and the coarticulation phase only from the motion information. That is why, few vision-based approaches for estimating coarticulation have been reported in the literature to date. The methods proposed so far do not address the problems encountered in recognizing all types of continuous hand gestures in a vision-based platform. Most of the existing algorithms use the glove environment and are successful only for some specific gesture vocabularies. The techniques developed so far for coarticulation detection are generally not useful for a broad gesture vocabulary. This is because, motion interpretation itself is an ill-posed problem in the sense that a unique solution cannot be guaranteed. Moreover, the existing algorithms do not address the problems associated with the extraction of the smoothed motion trajectory and the more consistent motion chain code.

This work focuses primarily on solving two basic problems in continuous gesture recognition, namely, gesture spotting and movement epenthesis detection, by extracting a smoothed motion trajectory via a modified motion chain

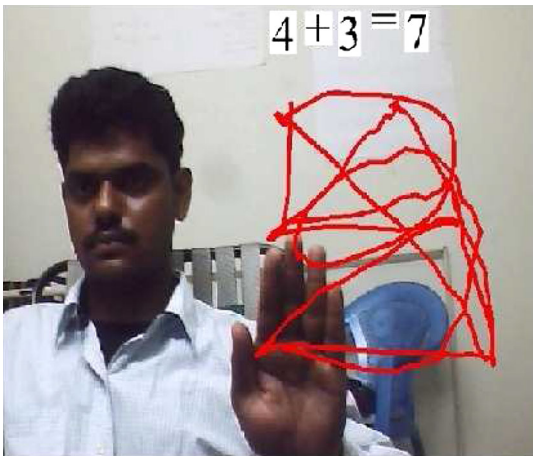

Fig. 13 A gesture-based calculator

code. Its main contribution is the extraction of three novel trajectory-based features to tackle movement epenthesis in continuous hand gesture recognition. The proposed system for movement epenthesis detection and subsequent recognition of individual gestures in a continuous stream of gestures promises to perform well on different types of gesture sequences having different spatiotemporal characteristics and motion behavior.

The proposed gesture recognition scheme overcomes the fundamental problems associated with continuous hand gesture recognition with the help of novel trajectory-based features. The experiments demonstrate that the system is 96.0 % accurate for recognizing isolated gestures, and 88.9 % accurate for continuous gestures. The number of thresholds to obtain the features L_E and P_H are application dependent. Evidently, CRFs can model the sequential data by considering the arbitrary dependencies, but fail to model the long-range dependencies. This is the only reason the recognition rate for gesture *One* is comparatively low. There is a probabilistic model called hidden conditional random fields (HCRF), which can model the long-range dependencies of the observed data [14]. Extending the proposed system using HCRF to recognize sign language is left for future work. Future work also includes the accurate segmentation and tracking of a moving hand from a cluttered dynamic background and the recognition of more fluent hand gestures in the form of continuous sign language.

References

1. Segouat J, Braffort A (2010) Toward modeling sign language coarticulation. Lecture Notes in Computer Science (Gesture in Embodied Communication and Human–Computer Interaction), vol 5934, pp 325–336
2. Yang R, Sarkar S (2006) Detecting coarticulation in sign language using conditional random fields. In: 18th international conference on pattern recognition (ICPR), vol 2, pp 108–112
3. Eisenstein J, Davis R (2005) Gestural Cues for Sentence Segmentation. Technical report, MIT AI Memo, pp 1–14
4. Lee HK, Kim JH (1999) An HMM-based threshold model approach for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(2):961–972
5. Bhuyan MK, Bora PK, Ghosh D (2011) An integrated approach to the recognition of a wide class of continuous hand gestures. Int J Pattern Recognit Artif Intell 25(2):227–252
6. Yang H-D, Sclaroff S, Lee SW (2009) Sign language spotting with a threshold model based on conditional random fields. IEEE Trans Pattern Anal Mach Intell 31(7):1264–1277
7. Alon J, Athitsos V, Quan Y, Sclaroff S (2009) A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans Pattern Anal Mach Intell 31(9):1685–1699
8. Zaki MM, Shaheen IS (2011) Sign language recognition using a combination of new vision-based features. Pattern Recognit Lett 32(4):572–577
9. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of 18th international conference on machine learning, pp 282–289
10. Wallach HM (2004) Conditional random fields: an introduction. University of Pennsylvania
11. Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. In: Proceedings of the IEEE, vol 77, no 2, pp 257–286
12. Chai D, Ngan KN (1999) Face segmentation using skin color map in videophone applications. IEEE Trans Circuits Syst. Video Technol. 9(4):551–564
13. Fitzgibbon A, Pilu M, Fisher RB (1999) Direct least square fitting of ellipses. IEEE Trans. Pattern Anal. Mach. Intell. 21(5):476–480
14. Quattoni A, Wang S, Morency LP, Collins M, Darrell T (2007) Hidden conditional random fields. IEEE Trans. Pattern Anal. Mach. Intell. 29(10):1848–1852