World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:2, No:9, 2008

Trajectory Guided Recognition of Hand Gestures having only Global Motions

M.K. Bhuyan, P.K. Bora, and D. Ghosh

***Abstract*—One very interesting field of research in Pattern Recognition that has gained much attention in recent times is Gesture Recognition. In this paper, we consider a form of dynamic hand gestures that are characterized by total movement of the hand (arm) in space. For these types of gestures, the shape of the hand (palm) during gesturing does not bear any significance. In our work, we propose a model-based method for tracking hand motion in space, thereby estimating the hand motion trajectory. We employ the dynamic time warping (DTW) algorithm for time alignment and normalization of spatio-temporal variations that exist among samples belonging to the same gesture class. During training, one template trajectory and one prototype feature vector are generated for every gesture class. Features used in our work include some static and dynamic motion trajectory features. Recognition is accomplished in two stages. In the first stage, all unlikely gesture classes are eliminated by comparing the input gesture trajectory to all the template trajectories. In the next stage, feature vector extracted from the input gesture is compared to all the class prototype feature vectors using a distance classifier. Experimental results demonstrate that our proposed trajectory estimator and classifier is suitable for Human Computer Interaction (HCI) platform.

***Keywords*—Hand gesture; Human Computer Interaction; Key video object plane; Dynamic time warping.

I. Introduction

The use of human hand as a natural interface for human-computer interaction (HCI) serves as the motivation for research in hand gesture recognition. Gestures provide an attractive and user-friendly alternative to interface devices like keyboard, mouse and joysticks in Human Computer Interaction (HCI) [1]. Vision-based hand gesture recognition involves visual analysis of hand shape, position and/or movement. One notable advantage of vision-based system is that it is easy to deploy and can be used anywhere in the field of view of the camera. The operator does not need to master special hardware like hand gloves or keyboard. Vision also allows a variety of gestures to be used as it is basically software based. Hand gestures can be described in terms of the following four major attributes:

- Hand configuration (i.e., posture) as defined by the flex angles of the fingers.
- Palm orientation.
- Hand movement as defined by the motion of the palm in space.

While static hand gestures are modelled in terms of hand configuration and palm orientation, dynamic hand gestures are described by hand trajectories and orientation in addition to temporally varying hand configuration and palm orientation. So, appropriate interpretation of dynamic gestures on the basis of hand movement in addition to shape and position is necessary. Another form of dynamic hand gesture that is in common use is in which the 2D gesture trajectory alone builds up a particular message. For these types of gestures, as the hand moves in space, there is not much change in the hand shape, finger configuration and palm orientation. That means, in these types of hand gestures there is global motion of the hand as a whole, while there is no partwise local movement in the hand. Examples of few such gestures are shown in Fig. 1. Dynamic hand gestures of this kind may be interpreted as a set of points in a spatio-temporal space represented by $\{(x_1, y_1), (x_2, y_2), ...)\}$, where (x_i, y_i) is the i^{th} instant spatial position of a predefined reference point on the hand. This is illustrated in Fig. 2 that shows the hand trajectory in the spatio-temporal space, where the spatial position of the hand in every frame along the temporal (time) axis is projected onto the xy-plane. Therefore, an essential step for recognizing such a gesture is to estimate the 2D gesture trajectory. Eventually, the gesture can be recognized by identifying the trajectory via analysis of some selected hand motion parameters.

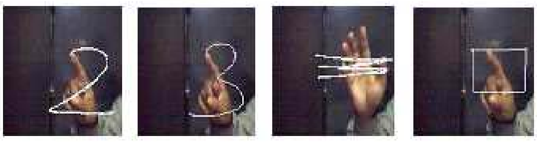


Fig. 1. Gestures representing “two”, “three”, “wavy hand” and “square” respectively.

As mentioned above, the first task in recognizing gestures having global motion is to track hand motion from the input gesture video sequence and subsequently estimate the trajectory along which the hand moves during gesturing. While trajectory estimation is quite simple and straightforward in glove-based hand gesture recognition system that provides spatial information directly, trajectory estimation in vision-based system may require applying complex algorithms to track hand positions across frames using image silhouettes and edges [2],[3]. One such method is given by [4] that uses skin colour to segment out hand regions and subsequently determines 2D motion trajectory. However, hand tracking

M.K. Bhuyan is with School of Information Technology and Electrical Engineering, University of Queensland, Brisbane Qld 4072, Australia, (Email : manas_kb@hotmail.com).

P.K. Bora is with Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, North Guwahati-781039, India, (Email : prabin@iitg.ernet.in).

D. Ghosh is with Faculty of Engineering and Technology, Multimedia University, Melaka Campus, Malaysia, (Email : debashis@mmu.edu.my).

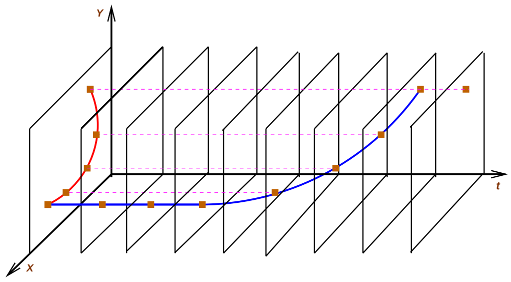


Fig. 2. 2D hand motion trajectory in spatio-temporal space.

algorithms in general are plagued by some difficulties such as large variation in skin tone, unknown lighting conditions and dynamic scenes.

For simple and small motion, the Kalman filter may be employed to estimate, interpolate and predict the motion parameters. But, this requires fine initial model and also demands additional computations whenever there is any rotation and change in hand shape. Also, the Kalman tracker is very much sensitive to background noise and is based on small motion assumption that often fails to hold in hand gesture motion [5]. Moreover, the Gaussian assumption in case of Kalman filtering is usually invalid in real world situation.

In [6], Black and Jepson extended the Condensation (conditional density propagation) algorithm to determine temporal trajectories . Since a sampling technique is used to represent the probability density in the condensation algorithm, their approach avoids some difficulties associated with Kalman filtering. Motivated by this, Black *et al.* used the mixed state condensation algorithm to recognize a large number of gestures using temporal trajectories [7]. Here, the authors focused solely on applying condensation to identify temporal trajectories, and left for future work the problem of leveraging the temporal models as constraints on object tracking. As such, they ignored the tracking aspect of the problem all together; only simple phicons (physical object icons) of distinctive colours are tracked by a colour blob tracker. Developing temporal trajectories using the condensation algorithm is a relatively ad-hoc process that may have difficulties in capturing the way in which different individuals make the same gesture. Moreover, in this approach all the trajectory points are considered for building up a trajectory model as well as for recognition. This seems to be computationally inefficient.

In an attempt to overcome all the above mentioned difficulties, we propose a model-based method for tracking hand motion in a gesture sequence. In our work, we use the concept of object-based video abstraction via segmenting the frames into video object planes (VOPs) with the hand as the video object. Next, a binary image for the moving hand is created and is used for tracking in subsequent frames using the Hausdorff tracker [8]. Next step of the proposed scheme is to select few key VOPs. Following this, we employ a centroid based method for estimating the hand trajectory, where centroids of the key VOPs are calculated on the basis of moment equations as well as from respective motion vectors. The computed centroids are then joined in the temporal sequence to obtain the final estimated trajectory. This is followed by final smoothing of the extracted trajectory and subsequently approximated the trajectory by applying MPEG-7 motion trajectory representation technique [9].

For pattern classification purpose, we propose to compare an input trajectory pattern with all predefined prototype gesture trajectory patterns using Dynamic Time Warping (DTW) [10]. The DTW is a template-based dynamic-programming matching technique that is applied to problems with temporal variability. It provides for nonlinear time alignment and normalization thereby eliminating the temporal variability that exists between different trajectories. Further, in the proposed pattern matching technique, instead of considering all the trajectory points, we map some selected points (key points) on the template trajectories onto the test trajectory. This discrete trajectory matching technique greatly reduces the computational burden.

The next important step in gesture recognition is to select suitable and appropriate motion features that describe the extracted input trajectory in terms of numerical values. Since hand gestures generally show large variations in shape, motion and textures, selecting good features is crucial for accurate recognition. In view of the present problem, we propose to use some basic trajectory features like the trajectory length, the number of significant curves in the trajectory and the velocity features so as to accomplish trajectory guided recognition successfully. Velocity features include the mean and the standard deviation of the speed by which the hand moves during gesturing, and the number of minima in the velocity profile that corresponds to the number of sharp corners in the trajectory. While the trajectory length and number of corners may be considered as static features, velocity features are the dynamic features considered for defining the nature of the hand motion. Finally, trajectory classification is accomplished on the basis of the Euclidean distance between the set of static and dynamic features extracted from the input and each of the prototype gesture trajectories contained in the gesture vocabulary database.

In a nutshell, the proposed method estimates hand motion trajectory from the segmented out VOPs which preserve the spatio-temporal information about the input hand gesture. Recognition of gestures is then accomplished via matching of trajectory shape and trajectory features. For fast recognition, we select few key points on the gesture trajectory and use only these key points for trajectory matching as well as for feature extraction. Key-point based trajectory representation significantly reduces memory requirement for storing the trajectory information. In the proposed method, spatio-temporal variation is collectively coped with the DTW-based nonlinear time alignment of trajectories and the maximum boundary deviation (*MBD*) - based shape similarity measurement. Once a particular gesture has been recognized, it may be mapped to a corresponding action for human-computer interaction. From application point of view, the proposed system is suitable for most Human Computer Interactive systems. The proposed scheme for hand gesture representation and subsequent recognition is described in the section to follow.

II. Proposed Trajectory Estimation Algorithm

A. Hand tracking and estimation of the motion vector

One obvious and essential step in hand motion trajectory estimation is to determine the motion of the gesturing hand from one frame to the next in the sequence. For this, we employ a hand tracking algorithm that is based on an object tracker that matches a two-dimensional binary image of the object (hand) in subsequent frames using the Hausdorff distance [8]. In our proposed technique, VOPs for different hand positions are obtained from the input gesture video sequence, where the hand is considered as the video object. In the next step, the Hausdorff object tracker finds the position where the input hand image best matches the next edge image and returns the motion vector that represents the best translation. The best match found indicates the translation the object has undergone, and the model is updated with every frame to accommodate for rotation and change in shape.

In the hand tracking process, we use a simplified Hausdorff tracker with reduced computational load. Considering the VOPs O and I, our aim is to find the motion vector between O and I through the Hausdorff tracker. An effective way to approximate the Hausdorff distance is the generalized Hausdorff distance in which the distances $min_{i_l \in I} \parallel o_k - i_l \parallel$ for all $o_k \in O$ are sorted in ascending order and the p^{th} value is chosen as $h_p(O,I)$. Similarly, $h_j(I,O)$ is defined as the j^{th} value of the ordered distances $min_{o_k \in O} \parallel i_l - o_k \parallel$ for all $i_l \in I$. The maximum of these two gives the generalized Hausdorff distance $H(O,I)$. In order to speed up the computation, the generalized Hausdorff distance is calculated via distance transform [11] which in turn is accomplished using the Chamfer $5-7-11$ mask. Further reduction in computation is achieved by searching the shifted hand image only at pixel locations in a search window around the model image in subsequent VOPs. The algorithm for determination of motion vector MV_i for the i^{th} VOP relative to the $(i-1)^{th}$ VOP may accordingly be described as given below.

Algorithm 3.1: Estimation of Motion vector

```
Given $(i-1)^{th}$ VOP $O$ and the $i^{th}$ VOP $I$, a set
    of translated vectors $\mathbf{T}$ and integers $p$ and $j$
begin
  for $\vec{t} = (t_x, t_y) \in \mathbf{T}$
      Calculate distance transform of edge images $O$ and $I$.
      Calculate $h_{p,t}(O,I)$.
      Calculate $h_{j,t}(I,O)$.
      Determine $H_t(O,I) = max\{h_p(O,I), h_j(I,O)\}$.
 end.
 Find $min\{H_t(O,I)\}$ over $\vec{t} \in \mathbf{T}$.
     Note the translation vector $t^{'} = (t^{'}_x, t^{'}_y)$ corresponding
     to $min\{H(O,I)\}$.
     $MV_i = (t^{'}_x, t^{'}_y)$.
 return $MV_i$
 end
```

B. Determination of the hand image centroid

In order to form the trajectory from the positions of the hand across the gesture sequence, it is necessary to define a single point on the hand and the trajectory can be obtained by joining the coordinate of this reference point in every frame in the sequence. For this, we choose to use the centroid of the hand as the reference point. This is because even if there is some local motion in the fingers, the centroid will be approximately at the center of the palm and hence the coordinate of this centroid in a frame will best represent the position of the hand in that frame.

The centroid of the hand in a frame can be easily computed from the first moment of the binary image of the hand. The VOPs obtained are binarized to form binary alpha planes by assigning "**1**" to the hand-pixels in a VOP and "**0**" to the background pixels. After determining the binary alpha plane corresponding to each VOP, the centroid of the hand in each VOP in the gesture sequence is determined using 0^{th} and 1^{st} order moments. The 0^{th} and the 1^{st} moments are defined as [12]:

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

$$M_{01} = \sum_x \sum_y yI(x,y) \qquad (1)$$

The centroid $c = (x_c, y_c)$ is calculated as

$$x_{\rm c} = \frac{M_{10}}{M_{00}} \qquad \text{and} \qquad y_{\rm c} = \frac{M_{01}}{M_{00}} \qquad (2)$$

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

Once the centroid of the hand in the first VOP is known, the centroids in all subsequent VOPs can also be determined using the motion information obtained above. The centroid in a VOP, say the i^{th} VOP, is determined by shifting the centroid of the previous VOP by the motion vector MV_i corresponding to the current frame. However, slight changes in the shape of the hand in successive frames may cause shifting of actual centroids from that computed from the motion vectors, as shown in Fig. 3. In view of this, an estimate for the centroid c_i of the i^{th} VOP is obtained by taking the average of the two centroid values computed from the moments and the motion vector respectively. The averaging is done to reduce the variance of the estimation error. This is illustrated in Fig. 4.

C. Trajectory formation and smoothing of the final trajectory

The final gesture trajectory is formed by joining all the calculated centroids in sequential manner while preserving the temporal information. However, the trajectory may be noisy due to the following reasons:

- Centroids may be far away from the actual trajectory due to change in hand shape.

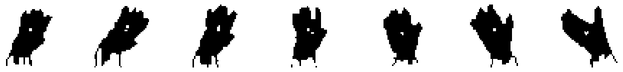


Fig. 3. Centroids calculated by motion vector and moment equations in VOPs.

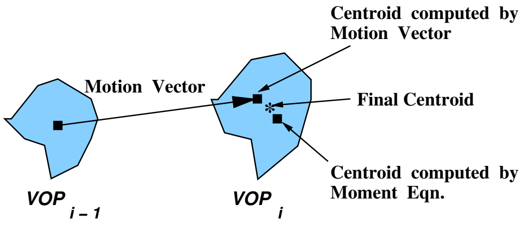


Fig. 4. Pictorial illustration for estimation of centroids in VOPs.

- Hand trembling and unintentional movements during gesturing may cause deviation of the centroids.
- All real gesture trajectories are generally open in practice as it is quite unlikely that in the case of closed gestures a signer will be able to complete a trajectory exactly at the point from where he has started. So, the unclosed end points may result in an open trajectory even if the actual trajectory is a closed one.

Therefore, in order to take care of this, the final trajectory is smoothened out by removing isolated noisy points with thresholding and replacing each point by the mean of the specified point and its two neighbours. Thus (x_t, y_t) is replaced by

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

To tackle the third point, the distance between two end points is computed. If it lies within a threshold, the two end points are merged.

D. Key VOP based trajectory estimation

The computational burden in estimating the hand trajectory may be significantly reduced by selecting the key video frames in the input gesture video sequence. Since during gesturing the human hand generally does not move very fast, the position of the hand in space does not change much from one frame to the next. Moreover, if we consider trajectory points from all the frames, the estimation error in any frame will change the trajectory. The trajectory will follow the estimation error and may not be smooth as a result. So the lines, curves *etc.*, may not be smooth and a somewhat jerky trajectory may be obtained. For example, a square may have sides as if they are serrated. Instead, if we consider trajectory points only from some key frames and subsequently interpolate the curve/line in between, the resulting trajectory will have a smoother appearance.

Therefore, for trajectory formation it may not be necessary to consider all the frames in a gesture sequence. Accordingly, we propose to select few key VOPs determined for the sequence. If the hand moves by a very small amount (both locally and globally) from one frame to the next, then the corresponding VOPs will be more or less alike. Based on this, the key VOPs in a gesture sequence are identified using the Hausdorff distance measure. Starting with the first VOP as the first key VOP, for every key VOP, the next key VOP is selected as the one for which this measure exceeds a predefined threshold [13]. After getting all the key VOPs in a sequence, hand trajectories are obtained by following the same procedure as above, but considering the key VOPs only.

E. Template trajectory generation

For trajectory guided dynamic hand gesture recognition, it is necessary to generate a prototype trajectory defining each gesture in the vocabulary to which an input gesture trajectory is to be compared for recognition purpose. For this, a sufficient number of training samples corresponding to a particular gesture are collected from a number of different signers, each signer making the same gesture at different times and under different conditions. The trajectories obtained from all the training sequences are aligned using the DTW technique, as described later in Section 4.3, and the mean trajectory is derived. This forms the prototype template trajectory corresponding to the gesture sign under consideration.

F. Key trajectory point selection on the template trajectory

All the template trajectories obtained above, one trajectory per gesture in the vocabulary, form the gesture database to which an input test gesture is to be matched during recognition. Each trajectory is stored in the form of an ordered set of coordinates $\{(x_1, y_1), (x_2, y_2), ...\}$, where (x_i, y_i) is the spatial position of the hand centroid in the i^{th} VOP (key VOP). That means, the number of points in the set equals to the number of VOPs (key VOPs). However, it may not be always necessary to store all the trajectory points. For example, a square gesture trajectory can be conveniently described by just the four corner points. In view of this, we propose to extract few key trajectory points to be stored in the database. This greatly reduces the memory requirement as well as speeds up the trajectory matching process during recognition.

Each key point on the trajectory is defined by its coordinate (x_i, y_i) in 2D space and the corresponding time instant t_i. These key points represent the prominent locations of the hand in the gesture trajectory. The number of key points is chosen by the user in such a way that the global precision and compactness required for the application is met. Variable time intervals between the key points are chosen to match the local trajectory's smoothness. The basic principle behind key trajectory point selection is merging of adjacent approximation intervals in the estimated template trajectory, until the interpolation error exceeds a predefined threshold, as used in MPEG-7 motion trajectory approximation [9]. We applied piecewise approximation of spatial positions of the centroids in successive VOPs following the motion trajectory approximation and representation technique as described below.

Trajectory approximation

- First order approximation:

$$x(t) = x_i + v_i(t - t_i), \text{where } \ v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{4}$$

- Second order approximation:

$$x(t) = x_i + v_i(t - t_i) + \tfrac{1}{2}a_i(t - t_i)^2 \quad , \text{ where}$$

$$v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \frac{1}{2}a_i(t_{i+1} - t_i) \tag{5}$$

The value of $y(t)$ may also be approximated in a similar manner. Here v_i and a_i represent hand velocity and acceleration respectively and are considered to be constant over the time interval $[t_i,\ t_{i+1}]$. The coordinate pairs (x_i, y_i) and (x_{i+1}, y_{i+1}) are the hand positions at times t_i and t_{i+1}.

We applied the sequential algorithm for selection of key trajectory points. The algorithm starts with an approximation interval, which is the union of the first two intervals. It sequentially adds the following points to expand this interval, until the interpolation error exceeds a given threshold. The process repeats for every three successive points in the trajectory path. The end points of each expanded interval constitute the key trajectory points.

III. Feature Extraction from Template Trajectories

Trajectory guided gesture recognition requires matching/comparing of an input gesture trajectory to each of the prototype gesture trajectories contained in the database. This may be accomplished either by direct trajectory shape matching and/or by trajectory feature matching. In our recognition method, we propose to employ both shape matching as well as feature matching, as described in Section IV. For feature based trajectory matching, we propose to use some static and dynamic features extracted from a given gesture trajectory. Static features relate to the shape of the trajectory while the dynamic features relate to the nature of hand motion during gesturing. The static features considered in our work correspond to the total length of the extracted trajectory, location of key trajectory points and the orientation of hand in the gesture trajectory. The only motion information proposed for trajectory classification is velocity. However, we also consider the acceleration feature which may be used for movement epenthesis determination in continuous gestures. The choice of all these features is based on observations over a large number of gesture samples. We observed the nature of hand motion when gestures are performed by different signers and the above mentioned features are proposed after analyzing their behaviours. We define the static features as low level features while dynamic features are high level features. This is because, even if the static features correctly match during classification, *i.e.*, even if the shape features match, it may not represent actual hand trajectory until motion parameters are compared. Motion features represent the dynamics of hand motion in the gesture trajectory. Therefore, for correct and precise recognition, both low level and high level features are required to be considered during trajectory matching.

A. Static features

Static features are derived from the contour of the gesture trajectory in space. Therefore, these features are related to the shape of the gesture trajectory pattern. However, the overall size of the gesture trajectory may differ from one signer to another. Also, it is possible that different signers make gestures at different positions in the camera frame. This also applies to gestures performed by the same signer but at different times and under different conditions. Accordingly, for truthful recognition of gestures, the algorithm demands for features that are invariant to size and position, *i.e.*, scale and translation (ST) invariance is desired.

Translation invariance can be easily obtained by considering the positions of the trajectory points relative to one reference point defined with respect to the trajectory pattern. The reference point that we have chosen is the center of gravity defined as

$$\hat{x} = \frac{1}{N+1}\sum_{i=0}^{N} x_i \ \text{ and } \ \hat{y} = \frac{1}{N+1}\sum_{i=0}^{N} y_i \tag{6}$$

where (x_i, y_i) is the i^{th} key trajectory point. and $N + 1$ is the total number of key points.

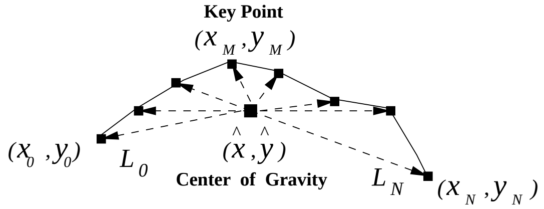


Fig. 5. Distances of the key trajectory points from the center of gravity.

For scale invariance, it is necessary to calculate the overall size of the gesture pattern in space and then normalize the extracted feature values with respect to the pattern size. This size is given by, as we propose, the average positional distance of all the key trajectory points from the center of gravity, calculated as

$$L_{avg} = \frac{1}{N+1}\sum_{i=0}^{N} L_i \tag{7}$$

where the distance of a key point from the center of gravity is simply the Euclidean distance between them, *i.e.*,

$$L_i = \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \tag{8}$$

This is depicted in Fig. 5. The static features that we propose to use are –

- Trajectory length, and
- Number of significant curves on the trajectory.

1) Trajectory length : For determining the trajectory length (*i.e.*, the total length traversed by the hand during a gesture), the sum of all the Euclidean distances between all pairs of consecutive key trajectory points is calculated.

$$l = \sum_{i=0}^{N} \{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2\}^{\frac{1}{2}} \tag{9}$$

where $N+1$ is the total number of key points on the trajectory. However, for a closed trajectory, the starting point (x_0, y_0) is same as the end point (x_N, y_N) and so only N number of distinct key trajectory points are available.

The trajectory length calculated above will vary with the size of the gesture pattern. So, in order to make it size-invariant, the size-normalized trajectory length is computed by dividing the calculated absolute trajectory length l by L_{avg} obtained in equation (7).

For a well-defined gesture pattern, this normalized trajectory length is generally fixed. For example, the normalized trajectory lengths in case of ideal gesture patterns representing "One", "Square" and "Circle" are 2, $4\sqrt{2}$ and 2π, respectively. However, real gestures generally do not follow the ideal trajectories but it is expected that this feature value will be near about the ideal one.

2) Number of significant curves on the trajectory: This feature is given by the number of key points on the gesture trajectory at which the curvature of the trajectory exceeds some predefined value giving rise to some sharp curves in the trajectory. That means, these are the points at which the hand changes the direction of motion by a significant amount. In order to extract this information, the direction of hand motion at every point on the trajectory is calculated as

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right), \ i = 1, 2, ..., N \tag{10}$$

The number of significant curves (N_θ) in the gesture trajectory is then obtained as the number of points at which the change in direction of hand movement exceeds some predefined threshold T_θ, i.e., $|\theta_{i+1} - \theta_i| \geq T_\theta$. From a large number of visualization experiments, it is observed that a human can generally perceive change in direction of hand movement only when the amount of angular displacement is approximately 45^o or more. Below this, confusion occurs in recognition. Accordingly, we select the threshold T_θ equal to 45^o.

B. Dynamic features

Motion features are computed from the spatial positions of the hand in the gesture trajectory and the time interval between two prominent hand positions. The two motion features, *viz.*, the velocity (v_i) and the acceleration (a_i) of the hand during gesturing at different positions in the trajectory are computed as given below.

$$v_i = \left\{\frac{x_{i+1} - x_i}{t_{i+1} - t_i}, \frac{y_{i+1} - y_i}{t_{i+1} - t_i}\right\}, i = 0, 1, 2, ...N-1 \tag{11}$$

$$a_i = \frac{v_{i+1} - v_i}{t_{i+1} - t_i}, i = 0, 1, 2, ...N-2 \tag{12}$$

In most situations, the velocity feature plays a decisive role during the gesture recognition phase. It is based on some of the important observations as listed below.

1) The magnitude of the velocity (speed) is generally more or less constant over a smooth trajectory, *e.g.*, gestures representing "One", "Circle", etc. On the other hand, large variation in speed is observed in trajectories with sharp changes in direction, *e.g.*, gestures representing "Square", "Wavy hand", etc.
2) Gestures with linear trajectories (straight line segments) are generally performed at approximately constant high speed, *e.g.*, the gesture representing "One". On the other hand, gestures with smooth non-linear trajectories (arcs, circles, etc.) are generally performed at approximately constant but medium speed, *e.g.*, gesture representing "Circle".
3) In gestures with sharp changes in trajectory direction, the speed of the hand generally becomes very low at points of such sharp changes. Ideally, the hand pauses (speed becomes zero) at these points. Accordingly, we have the following deductions:
    - The average speed is generally medium or low – the more the number of direction changes, the lower is the average speed.
    - The mean deviation of the speed from the average speed is generally large.
    - The speed of the hand becomes very low at points of sharp changes in the trajectory direction.

Based on the above deductions, we propose the following features for defining the dynamics of a gesture trajectory.

1) Average speed v_{avg} over the whole trajectory length.
2) Standard deviation σ_v of the speed over the whole trajectory length.
3) Number of minima $N_{v_{min}}$ in the velocity profile that have speed below some threshold $T_{v_{min}}$.

Note that for any gesture the hand always starts from a pause and ends in a pause state. To take care of this, the starting and ending points in the velocity profile are not considered in determining $N_{v_{min}}$. Also, the threshold $T_{v_{min}}$ is generally chosen very small so that for smooth trajectories the number $N_{v_{min}}$ is zero.

The above three features give the characteristics of the hand motion while forming different gesture patterns in space. For example, the trajectory for "One" gesture will give high v_{avg}, low σ_v and $N_{v_{min}} = 0$. Similarly, "Circle" will give medium v_{avg}, low σ_v and $N_{v_{min}} = 0$. On the other hand, a "Square" gesture will give medium or low v_{avg}, high σ_v and $N_{v_{min}} = 3$. Figure 6 shows the typical velocity profiles for gestures representing "Circle" and "Square" respectively.

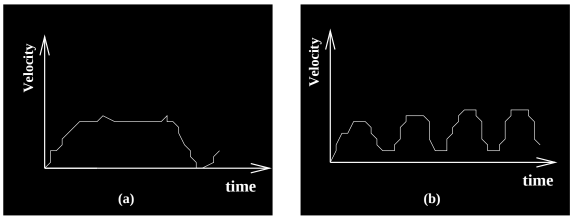


Fig. 6. Typical velocity plots of gestures representing (a) Circle and (b) Square.

From a number of experiments, we find that the acceleration feature does not play much significant role for trajectory classification/matching. However, this feature may be useful in discriminating movement epenthesis phase in a continuous dynamic gesture sequence. During continuous gesturing, one gesture follows another in a sequence. It is observed that during transition from one gesture to the next the hand generally moves with high acceleration from the end point of one gesture to the start position of the next gesture. However, movement epenthesis detection by using acceleration feature is beyond the scope of this paper.

C. Forming prototype feature vectors and knowledge-base for gesture matching

The set of feature values extracted from the template trajectory for a particular gesture, as described above, forms the prototype feature vector that gives the mathematical description for that class of gesture. As pointed out earlier, the selected features are trajectory length (l), number of significant curves on the trajectory (N_θ), average speed (v_{avg}), standard deviation of the speed (σ_v) and the number of minima in the velocity profile $(N_{v_{min}})$. Thus, the feature vector is given by

$$\mathbf{F} = \begin{bmatrix} \frac{l}{L_{avg}} \\ N_\theta \\ v_{avg} \\ \sigma_v \\ N_{v_{min}} \end{bmatrix} \tag{13}$$

On the other hand, different features may be weighted according to the degree of their importance. Thus the modified feature vector $\tilde{\mathbf{F}}$ having w_i weight value for i^{th} feature may be written as:

$$\tilde{\mathbf{F}} = \begin{bmatrix} w_1 \;\; \frac{l}{L_{avg}} \\ w_2 \;\; N_\theta \\ w_3 \;\; v_{avg} \\ w_4 \;\; \sigma_v \\ w_5 \;\; N_{v_{min}} \end{bmatrix} \tag{14}$$

Note that it is not always mandatory to consider all these features for classification. Depending on the nature of gesture trajectory different feature combinations and weights may be considered.

Finally, all prototype feature vectors, one per gesture class, together form the knowledge-base which is used for gesture matching during classification, as described in the next section.

IV. Proposed Method for Gesture Classification

A. Gesture trajectory estimation

For an input test gesture, the first step in recognition involves gesture trajectory estimation. The method employed for the purpose is same as that during training and as described in Section 2.3 above.

B. Shape based trajectory matching

As a first level of gesture classification, the shape of the gesture trajectory obtained above is compared with each of the template (prototype) trajectories in the knowledge-base. The measure used for trajectory shape matching is maximum boundary deviation (MBD) which is defined as the maximum of the distances of all pixels in the input trajectory to their nearest pixel in the template trajectory, expressed as

$$\text{MBD} = \max_{p_i \in P} \min_{q_k \in Q} \parallel p_i - q_k \parallel \tag{15}$$

where $\parallel . \parallel$ represents Euclidean distance, P represents the set of points on the input trajectory, and Q represents the set of points on the template trajectory. This is illustrated in Figure 7. Essentially, the MBD measure is the Hausdorff distance measure used for shape comparison.

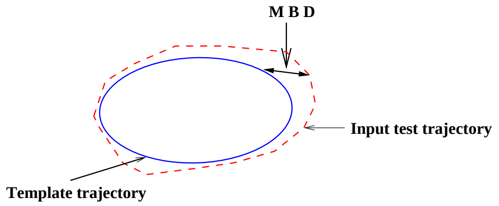


Fig. 7. Trajectory shape matching via maximum boundary deviation.

To allow for certain degree of shape variance in the matched trajectories, a threshold T_s is defined. Template trajectories that have MBD less than T_s with respect to the input trajectory are the candidate trajectories considered for matching in the next stage while all other template trajectories are rejected from further consideration. This reduces the overall computational load.

C. Nonlinear time alignment and normalization of motion trajectories by dynamic time warping

1) Dynamic time warping : Dynamic time warping (DTW) is a template-based dynamic matching technique widely used in several algorithms for speech recognition [10]. Even if the time scales of a test sequence and a reference sequence are inconsistent, DTW can still successfully establish matching as long as the time ordering constraints hold. Dynamic time warping is a method for computing a nonlinear time normalization between a template vector sequence and a test vector sequence. These two sequences may be of different lengths. The DTW algorithm, which is based on dynamic programming, computes the best nonlinear time normalization of a test sequence in order to match a template sequence by performing a search over the space of all allowed time normalizations. The space of all time normalizations allowed is judiciously constructed using certain temporal consistency constraints. Following is the list of all the temporal consistency constraints that we have used in the DTW implementation for motion trajectory normalization.

- *End point constraint*: The beginning and the end of each sequence should be rigidly fixed. For example, if the template sequence is of length I and the test sequence is of length J, then only those time normalizations that map the first point of the template to the first point of the test sequence and also map the I^{th} point of the template sequence to the J^{th} point of the test sequence are allowed.

- *Monotonicity constraint*: The warping function that maps the test sequence time to the template sequence time should be monotonically increasing. In other words, the sequence of "events" in both the template and the test sequences should be same.
- *Continuity constraint*: The warping function should be continuous.

Dynamic programming is used to efficiently compute the best warping function and the global warping error. Figure 8 demonstrates the matching technique in DTW where correspondence between points in two curves is determined.

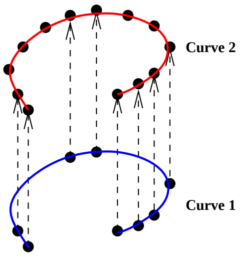


Fig. 8. Correspondence between points in two curves in DTW.

2) Time alignment and normalization by DTW and key point selection: The pattern-matching technique for gesture recognition needs to be able to compare sequences of motion features of the extracted trajectory. The problem associated with the comparison of motion features of a gesture trajectory arises from the fact that different motion features are seldom realized at the same speed across different instances of the same gesture. Hence, when comparing different tokens of the same gesture, variations due to difference in gesturing speed should not contribute to the dissimilarity score. Variation in speed and time of performing a particular gesture results in variation in length of the extracted trajectory. Thus, there is a need to normalize these fluctuations prior to the comparison of motion trajectory. The fundamental point is that finding the "best" alignment between a pair of patterns is functionally equivalent to finding the "best" path through a grid mapping the motion features of one pattern/trajectory to those of the other pattern. Finding this best path requires solving a minimization problem to evaluate the dissimilarity between the two motion trajectories. This may be accomplished by the DTW algorithm. The goal of the DTW algorithm is to find an optimal time-alignment between two patterns **R** (Reference) and **T** (Test) by evaluating various permitted pairings between the points of the two sequences, and selecting the best alignment path through these points based on some optimality criteria and search constraints [10],[14]. A basic version of the DTW algorithm is illustrated in Fig. 9. In this figure, the two axes represent the trajectory points in sequence, where the horizontal axis corresponds to the input test trajectory **T** and the vertical axis corresponds to the reference trajectory **R**. There are many possible pairings of these points subject to the constraints of monotonicity, continuity, boundary alignment, and search window width, as expressed by the following conditions on the arrays i and j of the indices of **T** and **R** respectively.

- Monotonicity condition: $i(k-1) \leq i(k)$ and $j(k-1) \leq j(k)$
- Continuity condition: $i(k) - i(k-1) \leq 1$ and $j(k) - j(k-1) \leq 1$
- Boundary condition: $i(1) = 1, \ \ j(1) = 1$ and $i(K) = I, \ \ j(K) = J$, where K is the final index and I and J are the total number of points in **T** and **R** respectively.
- Search window condition: $\mid i(k) - j(k) \mid \leq r$, where r is the maximum permitted search window width.
- Slope constraint condition: The path should not be too steep or too shallow. This prevents very short sequences matching with very long ones.

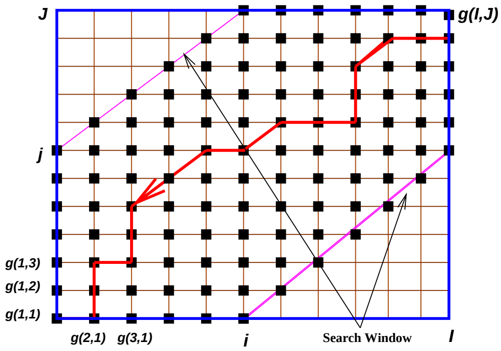


Fig. 9. Finding the optimal warping function using DTW.

The cumulative distance $g(i,j)$ between the two sequences from the beginning (origin of the grid) of the trajectories to point (i,j) in the grid is calculated as

$$g(i,j) = min \begin{bmatrix} g(i-1,j) + d(i,j) \\ g(i,j-1) + d(i,j) \\ g(i-1,j-1) + d(i,j) \end{bmatrix} \qquad (16)$$

Here $d(i,j)$ is the distance between the i^{th} point of the first gesture trajectory **R** and j^{th} point of the second trajectory **T** under consideration. The initialization of the above recursion is done by $g(1,1) = d(1,1)$. The overall distance or distortion between the two sequences is given by $D = g(I,J)$, where I and J are the total number of points on the two trajectories. A nonnegative function $w(k)$ is used to weigh $d(i(k),j(k))$ while finding the optimal path. This gives flexibility to the optimal warping path.

The optimal warping function or path is found recursively by starting at point (I,J) and backtracking to the beginning of the gesture trajectories. This is shown by the red coloured line segments in Fig. 9. The choice of the slope of the path and a weighting function $w(k)$ to weigh $d(i,j)$ in equation (16) is to be made for optimal warping [14]. For our purpose, we select slope equal to zero and a weighting function evaluated as

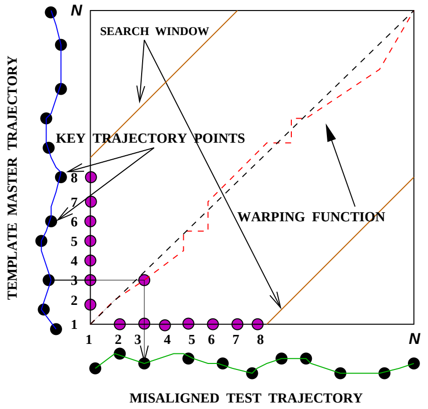


Fig. 10. Key point selection in the input trajectory by DTW matching.

$$w(k) = \{(i(k) - i(k-1)\} + \{(j(k) - j(k-1)\} \qquad (17)$$

The basic concept of the proposed trajectory matching under time normalization is shown in Fig. 10. Here the vertical and horizontal axes represent the spatial axes of a chosen template trajectory and that of the input candidate trajectory, respectively. The black dots on the template trajectory indicate the key points chosen for matching. The optimal warping function that establishes the best correspondence between the points of the input trajectory with the master template trajectory is indicated by the red coloured dotted line segments. The key points of the test trajectory are determined by finding points corresponding to the pre-calculated key points of the template trajectory through the optimal warping function.

D. Feature extraction and classification

In the final stage of classification, feature values are extracted from the time aligned (normalized) input gesture trajectory as described in Section III and the test feature vector is formed as given in equation 14.

The feature vector thus formed is compared to each of the prototype feature vectors of the candidate trajectories in the knowledge base except for those gesture trajectories that have been rejected in the first level of classification in the form of trajectory shape matching. The vectors are compared in terms of the Euclidean distance. The input pattern is classified to that gesture class for which the distance measure is minimum. However, if this minimum distance is greater than a threshold T_D then the input gesture is rejected from classification in the sense that the input gesture does not resemble any gesture in the vocabulary.

To estimate the threshold T_D, all the sample training trajectories are treated as the input trajectories. The minimum of the distances of the pairs of feature vectors for each gesture class is determined. Assuming that the training set does not contain any noisy gesture pattern, the maximum of all these minimum distances serves as a good measure of the threshold.

The minimum distance classifier may be alternatively implemented through the Mahalanobis distance. It is to be noted that calculation of Mahalanobis distance requires to know the

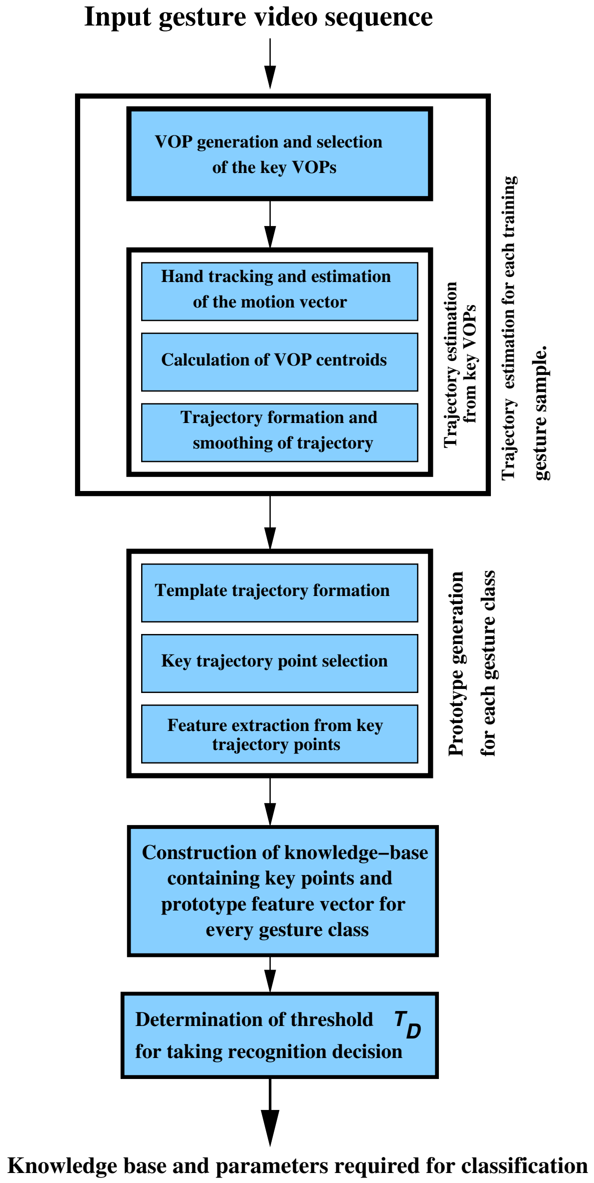

Input gesture video sequence
VOP generation and selection of the key VOPs
Hand tracking and estimation of the motion vector
Calculation of VOP centroids
Trajectory formation and smoothing of trajectory
Trajectory estimation from key VOPs
Trajectory estimation for each training gesture sample.
Template trajectory formation
Key trajectory point selection
Feature extraction from key trajectory points
Prototype generation for each gesture class
Construction of knowledge–base containing key points and prototype feature vector for every gesture class
Determination of threshold T_D for taking recognition decision
Knowledge base and parameters required for classification

Fig. 11. Proposed training algorithm.

covariance matrix describing the spread of the feature vectors for each gesture class. This may be estimated during the training phase demanding additional computation.

The flowchart showing the proposed training and classification algorithms for trajectory guided gesture recognition are shown in Fig. 11 and Fig. 12.

V. Experimental Results

In our experiment, we considered planar hand gestures generated in front of the camera. The gesture recognition system was implemented on a personal computer with an image capture board. The input images were captured by a CCD camera at a resolution of 120×160 pixels. The gesture videos were recorded at a normal frame rate (30 frames per second). A vocabulary of ten different gestures having only global motion was used. For their simplicity,

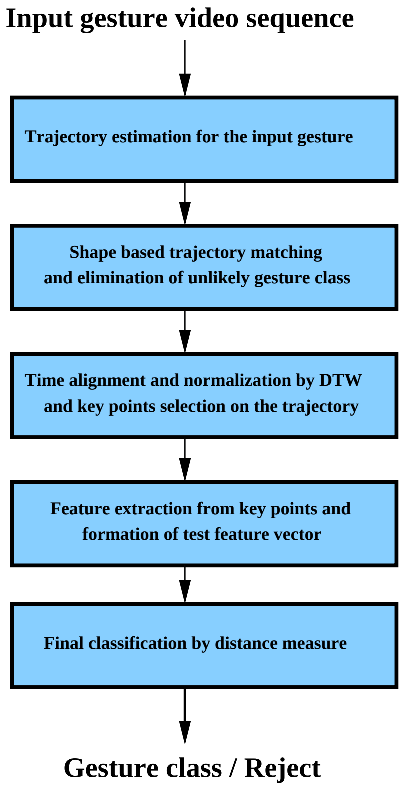

Input gesture video sequence
Trajectory estimation for the input gesture
Shape based trajectory matching and elimination of unlikely gesture class
Time alignment and normalization by DTW and key points selection on the trajectory
Feature extraction from key points and formation of test feature vector
Final classification by distance measure
Gesture class / Reject

Fig. 12. Proposed gesture classification algorithm.

these gestures may be suitable to several special applications like robot control and gesture based window menu activation in an HCI platform. They are the gestures showing “One”, “Two”, “Three”, “Five”, “Six”, “Seven”, “Alpha”, “Circle”, “Square”, “Circle” and “Diamond”, as illustrated in Fig. 13. All these gestures are single handed gestures. Each of them are free of intermediate movement epenthesis. The proposed algorithm was applied to a database containing 1500 gestures of these classes performed by 10 persons. Each person was asked to do his/her gestures 15 times for each of the 10 gestures. The gesture data were partitioned into two sets, one for training and the other for testing of the proposed algorithm. The partitioning of data is done as shown in Table I. The training and testing gesture data used for our experiment were naturally generated without any constraints on the speed or size of a gesture.

The extracted trajectories for six of these gestures, *viz*., “Circle”, “Square”, “One”, “Two”, “Seven” and “Six” are shown in Fig. 14. The first row shows the extracted trajectories and second row shows the trajectories after smoothing. Figure 15 shows the trajectories obtained using only the key VOPs. Visual analysis shows that the proposed trajectory estimator extract different gesture trajectories as desired for the present purpose. The smoothing operation removes some of the jerks in the gesture trajectories present before smoothing. Further, the gesture trajectories extracted from the key VOPs shows comparatively smoother trajectories. For example, the gesture trajectories “Two”, “Circle”, “Square” and “Six” extracted from key VOPs appear less noisy than those estimated from all the VOPs. These motion trajectories encode dynamic characteristics of hand gestures.

As discussed earlier, the shape based trajectory matching selects those trajectories which are similar in shapes with the test trajectory. The feature matching step matches features of the test trajectory to those of the trajectories selected during shape matching. In our experiment, we assigned equal weights of 1.0 to all the component of feature vector $\tilde{\mathbf{F}}$. Table II shows the classification accuracy in case of the selected gesture vocabulary. In all these cases the proposed recognizer gives recognition accuracy of 90.0% or more with an overall average accuracy of 95.80%, which may be considered as a good recognition rate for HCI applications. For the gesture trajectory “One”, v_{avg} is high and $N_{v_{min}} = 0$ and on this basis we achieved cent percent recognition accuracy for this gesture. It is seen that the proposed system gives reasonably good recognition accuracy for gesture trajectories having sharp corners, where the velocity of the hand drops to a minimum value. In all these cases, velocity based features shows good discriminative power in classification. On the other hand, for trajectories with minimum number of corner points, the dynamic features are less dominant. The low recognition rate of gesture trajectories “Circle” and “Six” may be attributed to this. Moreover, it is seen during experimentation that most errors come from the failure of hand extraction that distorts the hand trajectory data. This is because some images were blurry due to rapid hand movements, thus affecting the segmentation of the hand. So, the motion trajectories might not have been extracted correctly.

Recognition rate obtained in our experiments is comparable to most other recognition methods available in the literature. For example, a stroke-based composite HMM method has been used by Kim and Chien for recognizing 3D hand trajectory gestures with an average accuracy rate of 96.88% [15]. But, the method uses cybergloves. As discussed earlier, Yang *et al.* proposed a method for extraction of 2D motion trajectories based on skin colour and subsequent gesture recognition by TDNN. Recognition rate of 96.21% was obtained on test set of gestures [4].

Since, not many vision based approaches for trajectory guided recognition have been reported for the class of gestures considered in this work, our method promises to be a significant development in the field of recognition of dynamic hand gestures having only global motion.

VI. Concluding Remarks

A method for trajectory guided dynamic hand gesture recognition is presented in this paper. The advantage of the system lies in the ease of its use. The user does not need to wear a glove, neither there is the need for a uniform background. Trajectory estimated from the corresponding VOPs in a gesturing sequence bears spatial information regarding hand positions in dynamic gestures and is utilized in the gesture classification stage. Recognition is performed in two stages – unlikely gesture classes are eliminated through simple trajectory point matching in the first stage. This greatly enhances the overall

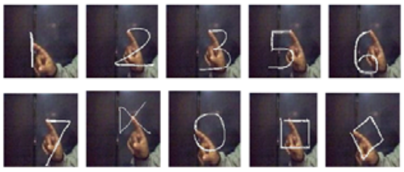


Fig. 13. Hand gestures showing different motion trajectories corresponding to “One”, “Two”, “Three”, “Five”, “Six”, “Seven”, “Alpha”, “Circle”, “Square” and “Diamond” respectively.

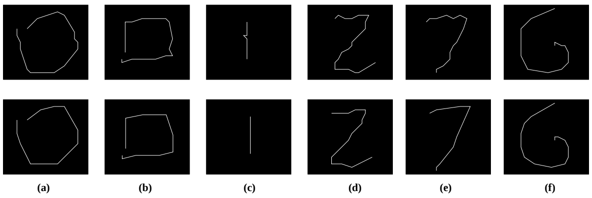


Fig. 14. Original extracted trajectories are shown in first row and smoothed trajectories for the same are shown in second row of the figure. The gestures represent (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Six, respectively.

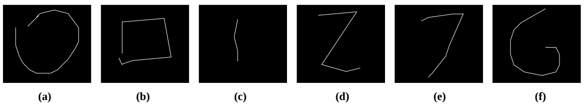


Fig. 15. Trajectories of (a) Circle (b) Square (c) One (d) Two (e) Seven and (f) Seven, respectively extracted from key VOPs.

speed of recognition. The second stage of recognition involves feature vector matching. The choice of features in the proposed algorithm is based on observations on the characteristics of real gestures. Experimental results confirm the appropriateness of these proposed trajectory features. However, one limitation of the proposed system is that the DTW has to align the test and prototype trajectories during each classification. This increases the computational load of classification for large vocabulary size.

REFERENCES

[1] V.I. Pavlovic, R. Sharma and T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: A review, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**(7) (1997) 677-695.

[2] D.L. Quam, Gesture recognition with a data glove, *Proc. IEEE Conf. National Aerospace and Electronics*, Vol. 2, 1990, pp. 755–760.

[3] D.J. Sturman and D. Zeltzer, A survey of glove-based input, *IEEE Computer Graphics and Applications*, **14** (1994) 30–39.

[4] M.H. Yang, N. Ahuja and M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(8) (2002) 1061–1074.

[5] Y. Wu and T.S. Huang, Hand modelling, analysis, and recognition for vision-based human computer interaction, *IEEE Signal Processing Magazine*, (2001) 51-60.

[6] M. Black and A. Jepson, Recognition temporal trajectories using the condensation algorithm, *Proc. International Conf. Automatic Face and Gesture Recognition*, 1998, pp. 16–21.

[7] M. J. Black and A. D. Jepson, A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions, *Proc. European Conf. Computer Vision*, Vol. 1, 1998, pp. 909-924.

[8] D.P. Huttenlocher, J.J. Noh and W.J. Rucklidge, Tracking non-rigid objects in complex scene, *Proc. 4th International Conf. Computer Vision*, 1993, pp. 93–101.

[9] B.S. Manjunath, P. Salembier, T. Sikora, (Ed.), *Intoduction to MPEG-7, Multimedia Content Description Interface*, (John Wiley and Sons Ltd, New York, 2002).

[10] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, (Prentice Hall, Englewood Cliffs, N.J., 1993).

[11] G. Borgefors, Distance transformations in digital images, C*omputer Vision, Graphics and Image Processing*, **34** (1986) 344–371.

[12] A. K. Jain, *Fundamentals of Digital Image Processing*, (Prentice-Hall, Englewood Cliffs, NJ, 1989).

[13] M.K. Bhuyan, D. Ghosh and P.K. Bora, Finite state representation of hand gestures using key video object plane, *Proc. IEEE Region 10 – Asia-Pacific Conf. TENCON*, 2004, pp. 21–24.

[14] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. on Acoustics, Speech and Signal Processing. **26** (1) (1978) 43–49.

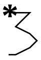
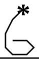
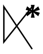
[15] I.C. Kim and S.I. Chien, Analysis of 3D hand trajectory gestures using stroke-based composite Hidden Markov Models, *Applied Intelligence*, **15** (2001) 131–143.

TABLE I
GESTURE PATTERNS FOR TRAINING AND TESTING.

Gesture	Meaning	Training Data	Test Data
	Circle	60	90
	Square	55	95
	Decision	55	95
	One	50	80
	Two	60	90
	Three	70	100
	Five	60	90
	Six	55	95
	Seven	60	90
	Alpha	60	90
Total		585	915

* indicates start point of a gesture.

TABLE II
EXPERIMENTAL RESULTS DEMONSTRATING THE PERCENTAGE OF ACCURACY IN GESTURE RECOGNITION USING OUR PROPOSED METHOD.

Gesture	Meaning	Test Data	Correct	Error	Reject	Recognition (%)
	Circle	90	81	7	2	90.0
	Square	95	92	3	0	96.8
	Decision	95	92	3	0	96.8
	One	80	80	0	0	100.0
	Two	90	88	1	1	97.8
	Three	100	92	2	6	92.0
	Five	90	88	0	2	97.8
	Six	95	87	6	2	91.6
	Seven	90	89	1	0	98.9
	Alpha	90	87	0	3	96.7
Total		915	876	23	16	95.84

* indicates start point of a gesture.