

Recognition of Dynamic Hand Gestures

Motilal Agrawal	Shyam Sadhwani	Santanu Chaudhury*	Subhashis Banerjee
Dept of CSE	Dept of CSE	Dept of EE	Dept of CSE
IIT Delhi	IIT Delhi	IIT Delhi	IIT Delhi
New Delhi-110016	New Delhi-110016	New Delhi-110016	New Delhi-110016
m1a@umiacs.umd.edu	shyam@delsoft.com	santanuc@ee.iitd.ernet.in	suban@cse.iitd.ernet.in

Abstract

This paper is concerned with the problem of recognition of dynamic hand gestures. We have considered gestures which are sequences of distinct hand poses. In these gestures hand poses can undergo affine transformations and discrete changes. However, continuous deformations of the hand shapes are not permitted. We have developed a recognition engine which can reliably recognize these gestures despite individual variations. The engine also has the ability to detect the start and end of gesture sequences in an automated fashion. The recognition strategy uses a combination of PCA based shape recognition with a HMM based temporal characterization scheme. A real time implementation of the scheme on standard hardware has been achieved. Experimental results establish the effectiveness of the approach.

Key words: Hand gesture, Hidden Markov Model, Contour Tracking, Real time system

1 Introduction

Interpretation of gestures is an important problem because gestures provide an attractive alternative to prevalent human-computer interaction modalities. In this paper we have focused on the problem of recognition of dynamic hand gestures. We have considered gestures which are sequences of distinct hand poses. In these gestures hand poses can undergo affine transformations and discrete changes. However, continuous deformations of the hand shape are not permitted. We have developed a recognition engine which can reliably recognize these type of gestures despite individual variations. The engine also has the ability to detect the start and end of gesture sequences in an automated fashion.

Pavlovic, Sharma and Huang [3] presents an extensive review of the existing techniques for interpretation of hand gestures. A large variety of techniques have been used for modeling the hand. All these may be

broadly classified into 3D hand model or appearance based model. Rehg and Kanade [4] designed a system called *Digit Eyes* which modeled the hand as 3D jointed cylinders. The second group of models is based on appearance of hand in images and hence are called appearance based models. Some are based on deformable 2D templates of the human hand [2]. Others such as [1] are based on tracking the 2D contours of the hand.

In this work we have developed a HMM based gesture recognition system which uses both temporal and shape characteristics of the gesture for recognition. The use of HMM for gesture recognition is not new. However, the methodology adopted for using temporal and shape characteristics in the HMM based approach is a new contribution of this work. We have used PCA based technique to extract shape descriptors for providing input to HMM. We have used a tracker for obtaining motion descriptors. The use of a quantization scheme has enabled us to obtain reliable descriptors which can accommodate local variations. We have developed a real time implementation of the recognition engine. Using application specific constraints for tracking and detection of the start and end of the gestures, we have been able to build a recognition system which can work at 20 frames/sec without specialized hardware.

2 Tracker Framework

In our gesture representation scheme, the hand has been considered as a smooth 2D curve. The moving hand has been tracked using a simplified form of the contour tracker developed by Blake et al. [1]. The tracker, in this case, consists of a Kalman filter based estimator for a piecewise smooth image plane curve in motion:

$$r(s, t) = ((x(s, t), y(s, t)))$$

The curve representation is in terms of B-Splines. Some advantages of the B-Spline curve representation

*Author for Correspondence

over the traditional snakes are local control, implied continuity, and compact representation.

$P(t)$, the position vectors along a B-Spline curve of order k and number of control points $n + 1$ is given by

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \quad t_{min} \leq t < t_{max}, \quad 2 \leq k \leq n+1$$

where the B_i are the position vectors of the $n + 1$ defining polygon vertices and the $N_{i,k}$ are the normalized B-spline basis functions given by the Cox-deBoor recursion formulae.

A moving hand, provided the fingers are not flexing, can be approximated as a planar rigid shape. When perspective effects are not too significant, a good approximation to the curve shape as it changes over time can be obtained by specifying Q , a linear vector valued function of the B-Spline coordinates (X, Y) . The Q parameters then specify the deformation (affine in our case) of the hand shape.

The relationships $Q \leftrightarrow (X, Y)$ between parameterizations are expressed in terms of two matrices M and W :

$$\begin{pmatrix} X \\ Y \end{pmatrix} = WQ + \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}$$

$$Q = M \left[\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} \right]$$

The matrices M and W are defined in terms of the shape template (\bar{X}, \bar{Y}) . In our case, since we assume hand to be a to be a planar shape and allow only translation and scaling, just four *affine* degrees of freedom (translations in z can be accounted by scaling) are required to describe the possible shapes of the curve. The space of possible Q -vectors is expressible as a four-dimensional linear subspace of Q -vectors, and a basis for this subspace is

$$\beta = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \bar{X} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \bar{Y} \end{pmatrix} \right\}$$

In that case the matrix M and W converting B-Spline control points $(X$ and $Y)$ to and from the four vector Q can be defined in the following way:

$$W = \begin{pmatrix} 1 & 0 & \bar{X} & 0 \\ 0 & 1 & 0 & \bar{Y} \end{pmatrix}$$

$$M = (W^t W)^{-1} W^t$$

2.1 Kalman Filter for Tracking

The control points to be tracked in an image specify the shape template which can undergo affine transformations specified by the matrix Q .

2.1.1 State Equation

The state equation for the motion has been considered as,

$$X_{n+1} - \bar{X} = A(X_n - \bar{X}) + \begin{pmatrix} 0 \\ W_n \end{pmatrix}$$

where

$$\bar{X} = \begin{pmatrix} \bar{Q} \\ \bar{Q} \end{pmatrix}$$

Therefore the equation of motion becomes

$$Q_{n+1} = A_0 Q_{n-1} + A_1 Q_n + (I - A_0 - A_1) \bar{Q} + B w_n$$

where, w_n represent the noise process and the matrix B couples the noise into dynamics.

For the case of uniform motion A_0 and A_1 become

$$A_0 = -I$$

$$A_1 = 2I$$

2.1.2 Measurement Model

After the prediction of contour by the Kalman filter, the measurement process finds the error in prediction. As B-spline curve is defined by a few control points, the measurement scheme works by adjusting the position of these control points. The control points are moved normal to the curve and the external energy of the curve is measured corresponding to each position of the control point. The external energy is given by $E = -|\Delta(I(x, y))|$. This corresponds to the gradient of the image and is an attractor towards edges. The new location of the control point is given by the point giving the lowest energy for the curve. The search span is limited to a distance of 10 pixels on either side. In order to measure the external energy of the curve, the curve is sampled at some fixed number of points. Also, since we have used a B-spline representation which has a local control, the external energy needs to be sampled only over a fixed number of spans on either side.

This process is repeated for each control point to get their new position. The Q vector for these control points is then computed. This is the error in prediction of Q by the Kalman filter; which is then multiplied by the Kalman gain to perform state updating.

2.2 Dynamic Hand Gestures

The contour tracker tracks only affine deformations of a fixed shape template. Therefore, this cannot be used in case of dynamic gestures which are composed of sequences in which hand can assume distinct forms. This requires a mechanism for segmentation of the

gesture into epochs consisting of similar hand shapes. The segmentation scheme is based on residual error of tracking. While tracking, we keep track of the external energy obtained during the measurement process. Abrupt change in the external energy indicates failure of the tracker. The point of losing the lock by the tracker is considered as the segmentation point.

3 Recognition of Hand Shape

In our work, we have considered a gesture to be characterized by the shape of the hand and the nature of motion of the hand. We have used appearance based model for representing the hand shapes. The shape representation process involves following steps:

1. localization of hand or bootstrapping
2. B-Spline Curve Fitting
3. PCA based modeling of hand templates

3.1 Localization of Hand

Initial shape template is extracted by separating the hand pixels from the background, tracing the outermost contour of the hand region and fitting a B-Spline curve through these traced points. Color cues have been used for extraction of hand pixels because of the characteristic color of human skin.

The traced contour points are first normalized to make it translation, scale and rotation invariant. They are made translation invariant by choosing the origin of coordinate system to be the mean of the traced points. Then the eigenvalues and the eigenvectors for the set of traced points are found. The two eigenvectors give the principal directions of variation. By setting the largest of the eigenvalues to a fixed value, we can make the template scale invariant. By choosing the two orthogonal eigenvectors as the coordinate axes we make the representation rotation invariant.

3.2 Recognition of Hand Shapes

After the normalization and B-Spline curve fitting we obtain control points for each hand shape (typically, 25 in our case). These control points are used for characterizing the shape templates. A shape template is represented by a 50 dimensional vector. An eigen-space is constructed using the training images of the hand shapes. An unknown shape template is classified using a minimum distance classifier.

4 Hidden Markov Model Based Recognition

Since gestures considered are dynamic processes, we need a mechanism to recognize gestures using their temporal characteristics. The recognition scheme must

be robust and should accommodate variations in the attributes of a gesture. We have adopted an HMM for this purpose.

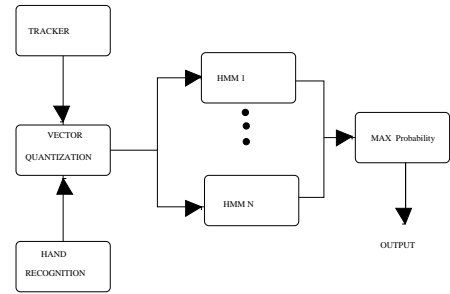


Figure 1: Different inputs and outputs to HMMs

Tracker provides temporal characteristics of the gesture. Output of the tracker is used for classifying the nature of the motion. Shape classification information is provided by the eigen-space based classifier. These symbolic descriptors are used as input for the HMM.

Major steps involved in recognition of gestures using HMM are :

1. Extraction of symbolic descriptors of the gesture at regular interval from the tracker and hand shape classifier
2. Training of HMMs by the sequence of symbolic descriptors corresponding to each of the gesture
3. For an unknown gesture find the model which gives maximum probability of occurrence of the observation sequence generated by the gesture.

For each gesture there is a HMM. Each HMM is trained by the symbol sequences obtained from the training set of each of the gesture.

4.1 Extraction of symbolic descriptors

The state, or “Q” matrix of a Kalman Filter has four elements which corresponds to four different transformations that hand can have i.e., x-translation, y-translation, x-scaling, y-scaling. If the hand is moving to (say +x direction) then only one component (corresponding to x-translation) of “Q” will be significant all other will be nearly equal to zero. Thus at each time instant we can have information about the motion by “Q” matrix. For generation of symbols (at time = t) difference of elements of “Q” matrix at time t and at start of gesture i.e., at time t = 0 is taken. Thus we have parameters available which corresponds to the position of hand with respect to start. The parameters which are positive assigned “+1”, negative ones correspond to “-1”, and those which have value near

to zero are taken as equal to zero. These temporal descriptors are generated after every 6 frames.

This symbol generation scheme can account for local variations of the motion and provide consistent set of descriptors. For example, consider a gesture in which the hand first goes to right, returns back and then goes to the left. In this gesture there are mainly two parts, one is when the hand is moving to right, and other when it is moving left beyond the original position. In the first part there is positive x-displacement and thus as hand moves to right only one of the component of “Q” matrix will be positive all other will be equal to zero most of the time. Same is the case when the hand is moving left from the original position. In this case there is negative x-displacement and the symbols generated are “-1” for x-displacement, zero for other motion parameters.

The hand shape classifier provides a unique identification code each shape class. This, combined with motion descriptors, provides the complete symbolic description of the image sequence. The shape classifier is invoked during initialization and at each sequence segmentation point.

5 Real Time Implementation

For real time implementation, the gesture recognition task was divided into two threads: grabber and tracker. Grabber grabbed images at rate of 25 frames per second and stored the images in buffer. The size of buffer was kept at 3 (i.e., 3 images can be kept). The tracker read images from this buffer and did tracking and recognition. The grabber and tracker operated as synchronized threads. Fig. 2 shows the processing scheme used. The implementation was done on SUN ultra SPARC workstation.

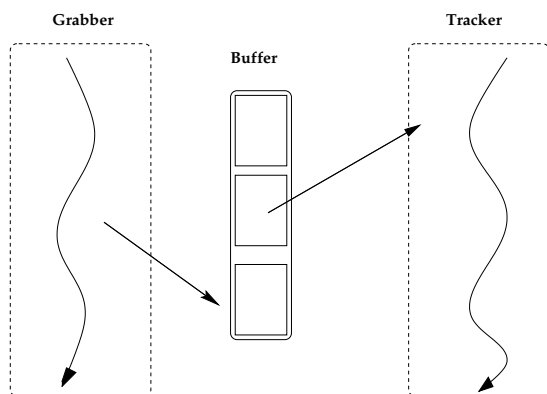


Figure 2: Grabbing and processing of images using threads

Timing analysis of the tracking-cum-recognition

process revealed that measurement module of the Kalman filter was causing the timing bottleneck. This was happening because finding minimum energy locations for the control points requires repeated calculation of the points on the B-Spline curve.

The equation for B-spline curve is given by

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \quad t_{min} \leq t \leq t_{max} \quad , 2 \leq k \leq n+1$$

where the B_i are the position vectors of $n+1$ control points and the $N_{i,k}$ are the normalized B-spline basis functions.

The value of $N_{i,k}(t)$ is independent of the position of the control points i.e., B_i . To reduce the computation we calculated the value of $N_{i,k}(t)$ for discrete values of t and stored it in an array. Now getting B-spline curve became a $O(n)$ function. Also in equation

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t)$$

$N_{i,k}(t)$ is nonzero only for 4 values of i (for order 4) so instead of summing it from $i = 1$ to $(n + 1)$ we are summing only for 4 values. All these optimizations improved performance of the tracker.

In the final implementation, we achieved a tracking rate of around 20 frames per second.

6 Application developed

For the purpose of experimentation we selected six different dynamic gestures. These gestures were logically associated with six different functions needed during presentation. The six different instructions given through gestures are :

1. Start presentation
2. Zoom in i.e., display current page with large scaling factor
3. Zoom out i.e., display current page with small scaling factor
4. Goto next page
5. Goto previous page
6. End of presentation

Gestures corresponding to above instructions are :

1. **Start** : At the start, the hand is closed and it moves towards the camera then it stops and hand shape changes from closed to open. Now this open hand moves back i.e., away from the camera. This complete gesture is for start of presentation.
2. **Zoom in** : For Zoom in, an open hand moves towards the camera , stops , and moves beyond the original position.

3. **Zoom out** : Zoom out gesture is reverse of the gesture for Zoom in. In this open hand first moves back i.e., away from camera , stops , and moves beyond the original position.
4. **Next Page** : For next page open hand moves to left side and then moves beyond the original position.
5. **Previous Page** : Gesture for previous page is reverse of the gesture for next page i.e., hand first moves towards right side and then moves beyond the original position.
6. **End** : Open hand moves towards the camera and then the shape of hand changes from open to closed and it moves beyond the original position.

In our application, camera is always grabbing images and is sending the image to “process_image ” function. The “process_image” function does recognition and tracking when user is delivering a gestural instruction, otherwise it just ignores the images. So now there are two problems

1. To decide when a user has started a gesture
2. To decide completion of a gesture

For the solution of Problem 1 , we made a small rectangle in the middle of the images grabbed by the camera. In each image this rectangular area is searched for the presence of hand colored pixels. If more than 75% of the pixels are of glove color then it assumes that gesture has started, otherwise tracker ignores the image. In other words, user is expected to bring his hand to the designated region for initiating a gestural action.

For the end of gesture problem, our solution is that whenever the user wants to end the gesture he can remove his hand away from the camera so that it does not appear in the designated region. When the hand is removed tracker sees significant change in energy and calls a function to detect the shape of hand. To find out the shape color segmentation is called and it detects no pixel is there corresponding to color of hand. Thus this is the end of the gesture. Now, the gesture is recognized using HMMs and related operation is performed.

7 Experimental Results

In this section we present the results of recognition of gestures obtained from different individuals. We assumed that user is gesticulating against a white background. We discuss results of each of the steps involved before presenting the overall results.

7.1 Results of Hand Detection

Colour cue based algorithm for detection of hand pixels produced correct results for 70 cases out of 100 experimental samples considered. The algorithm failed to extract proper hand contours always because of the different types of marking visible on the hand. In order to minimize this effect in subsequent experiments user was made to wear black gloves.

7.2 Results of Tracking

The tracker has been found to track shape template reliably at 20 frames/sec. Some of the tracking results are shown in fig. 3 (where hand moves towards camera) and fig. 4 (where hand move away from the camera). In fig. 5 we show a sequence where tracker detects a segmentation point at the intermediate pose.

We have found that tracker loses the track of hand under following conditions:

1. If hand moves very fast.
2. If hand moves away from the camera (affine model is invalid in this case).
3. If initialization is not good



Figure 3: Hand moving towards camera



Figure 4: Hand moving away from camera



Figure 5: Open hand few frames before closing, intermediate and finally closed

7.3 Results of Static Recognition

Here, we are illustrating results of hand shape recognition with reference to three static hand poses. Poses chosen for recognition are open hand, pointing finger, closed hand (fig. 6).

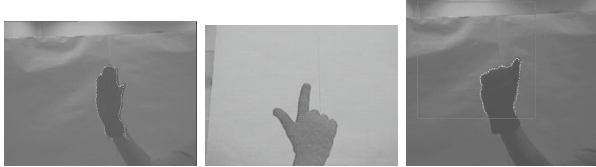


Figure 6: Open hand, Pointing finger, Closed hand

For each of these hand poses a training set of 25 was obtained from different individuals. The classifier was tested with 100 different samples. The recognition of the classes was correct in about 93 percent of the hand poses. Those samples which were not correctly recognized included those where the wrist was dominant in the traced contour.

7.4 Results of HMM based Recognition

The symbols were generated after every six frames and the HMM based modeling was tried out for different values of the number of states and the degree of the Left Right HMM. Experimentally the best recognition results were obtained when the number of states was taken as six and the out degree was three. In order to do an extensive study of the recognition rate different users were asked to perform the gestures and the number of correctly recognized gestures is provided in Table 1. HMM's were trained with the gesture sequences obtained from the first user.

User	Start	Zoom in	Zoom out	Previous	Next
1	24/25	22/25	21/25	23/25	21/25
2	9/10	8/10	9/10	9/10	8/10
3	7/10	6/10	8/10	7/10	7/10
4	7/10	7/10	7/10	6/10	7/10
5	8/10	7/10	8/10	7/10	8/10

Table showing recognition rate for different gestures

Confusion occurred in most of the cases in complimentary gestures viz. zoom in and zoom out, next and prev. In all the cases the duration of the gestures was different and hence our recognition scheme is not dependent on the length of the gesture. The results also suggest that the recognition scheme is not user dependent.

8 Conclusions

We have developed a robust system for real time recognition of dynamic hand gestures. The approach provides a new mechanism for fusing temporal and shape characteristics of the gestures for HMM based recognition. Experimental results have established effectiveness of our approach. Some of the key areas for further work are: dealing with non-uniform background, and extending the tracking for deformable contours so that initialization is not required every time the hand changes shape.

References

- [1] A. Blake, M. Isard, and D. Reynard. Learning to track the visual motion of contours. *Artificial Intelligence*, 1995.
- [2] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61, January 1995.
- [3] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.
- [4] J. M. Rehg and T. Kanade. Digteyes: Vision-based human hand tracking. Technical Report CMU-CS-93-220, 1993.