Dynamic Hand Gesture Recognition Using Predictive Eigen Tracker

Kaustubh S. Patwardhan Sumantra Dutta Roy
Department of Electrical Engineering, IIT Bombay, Powai, Mumbai - 400 076, INDIA
{kaustubh, sumantra}@ee.iitb.ac.in

Abstract

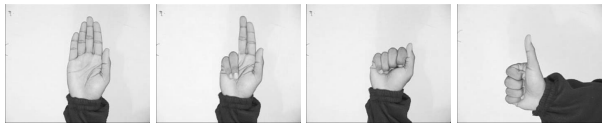
In this paper we present a novel framework to model a dynamic hand gesture by k-dimensional vector that incorporates both - the hand shape as well as the trajectory information. We introduce the notion of 'distance' between dynamic gestures to help choose a proper set of gestures for the gesture vocabulary. We also utilise inter-gesture distances for gesture recognition. We show encouraging results on a representative set of gestures selected according to the above criteria.

1. Introduction

In this paper, we propose a novel framework to model a gesture as a k-dimensional vector that combines the hand trajectory as well as hand shape information. This framework also allows us to properly choose the gesture vocabulary so as to maximize the recognition accuracy.

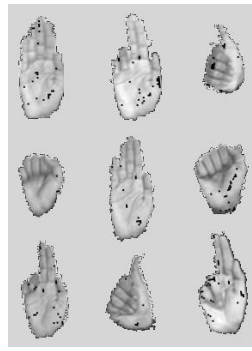
A dynamic hand gesture comprises a sequence of hand shapes with associated spatial transformation parameters (such as translation, rotation, scaling/depth variations etc.) that describe the hand trajectory. Gesture recognition schemes can be broadly classified into two groups. In the first approach, a gesture is modeled as a time sequence of states. Here, one uses Hidden Markov models (HMM), discrete finite state machines (DFA), and variants thereof for gesture recognition. In the second approach, one uses dynamic time warping to compensate for the speed variations (undulations in the temporal domain) that occur during gesticulation. Gesture recognition schemes can also be categorised on the basis of the parameters that are used to model the appearance of the hand e.g., hand silhouette-based model, graph-based model, use of Fourier descriptors, b-splines etc.

Pavlovic *et al.* [10] give an extensive review of the existing hand gesture recognition techniques. Nam *et al.* [8] extract the parameters of the non-linear arm motion, and unify them with shape attributes of the hand for recognition. A gesture is broken down into smaller *recognition units*, that are characterized by the arm motion in a single 2-D plane (in 3-D space). HMM-based framework is used for robust estimation the individual recognition units from the feature sequence. A sequence of recognition units is interpreted as a meaningful gesture. However, exact delimiting of recognition units is essential for good results. Kapuscinski [11] uses skin colour cues to extract the shape and orientation of the hand. This information is combined with the hand motion estimates, and analysed using a bank of HMMs to recognise the gesture performed. The system, however, is not very robust to background clutter, and structured noise. Min *et al.* [7] use coordinates of center of the detected hand region as features to estimate gestures. A Task-Specific state transition machine is used to detect and differentiate between static and dynamic gestures. Dynamic gestures are represented by a combination of Cartesian space features (e.g., vector velocity) and polar space features (distance from, and angle subtended with the center of the trajectory), and recognised using an HMM-based framework. Wah *et al.* [9] describe hand shape using normalized Fourier descriptors. A radial basis function network is used to map the observed hand shape to a set of five predefined shapes. This shape information along with motion information (of the centroid of the binary hand image) is given to an HMM bank to estimate the gesture. Yeasin *et al.* [13] extract temporal signature of hand motion. Laplacian of Gaussian (LoG) operator is used in temporal domain, to estimate motion break-points. Gabor like quadrature filters are used over portions of uniform motion, to extract the dominant motion component which is analysed in a DFA framework to estimate the gesture performed. However, structured background noise can adversely affect dominant motion extraction. Jerrah *et al.* [2] use Neuro-fuzzy systems for gesture recognition. The normalized lengths of vectors, running from centroid of detected hand region to hand region border near the finger-tips are used as features. 'Adaptive Neuro Fuzzy Inference Systems (ANFIS)' are used to process these features, and estimate the gesture performed. This requirement of visibility of the fingertips places restrictions on the gesture-set. Hongo *et al.* [6] describe a four camera system to track and recognise hand



(A) (B) (C) (D)

(a) Different hand shapes used in the gesture-set.



(b) Sample hand shapes detected by the tracker.

Figure 1. Various Hand Shapes

shapes and human faces. The system uses depth estimates to enhance the tracking of hands and face. Gestures are represented using directional features in sub-sampled images. A hierarchical linear discriminant analysis is carried out on these feature images to recognise the gestures. However, having multiple cameras is not always feasible. Ahmad *et al.* [1] present a Point Distribution Model (PDM)-based scheme for hand tracking and gesture recognition. Triesh *et al.* [12] present a system for automatic classification of hand postures using elastic graph matching. Hand postures are modeled by labeled graphs and an iterative algorithm is used to match the image with different graphs for shape estimation. However, this is a compute-intensive process. Zhu *et al.* [14] use geometric moments of hand region pixels to represent the shape of hand. Linear re-parameterization is used to combat variability in speed of gesticulation. Normalised correlation is used as measure of similarity between test gesture and template gestures.

In this paper, we use a predictive EigenTracker to track the gesticulating hand. EigenTracker [3] is an appearance-based tracker that can track objects simultaneously undergoing image motion and changes in appearance. In our earlier work [4], [5] we enhance the EigenTracker by augmenting it with a CONDENSATION-based predictive framework. Of great use is the ability of a predictive EigenTracker to learn and track unknown views of the object *on the fly*. The output of the tracker is a set of object reconstruction coefficients describing the view of the object and affine transformation coefficients that describe the object motion. This information is used to represent the gesture as vectors. The next section describes our gesture model in detail. Experimental results are presented in section 3. We

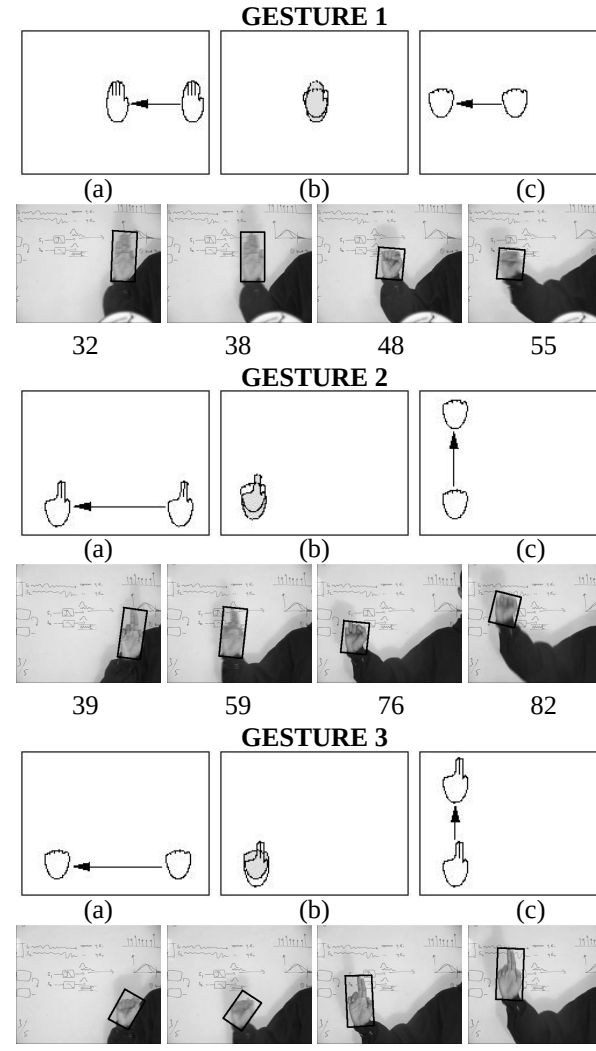


Figure 2. First three gestures in the vocabulary.

conclude in the last section and identify areas for further work.

2. Eigenspace Modeling: Shape, Trajectory

In this approach, the gesture is modeled by a k-dimensional vector. The components of this vector are the parameters that describe the different hand shapes and the portion of trajectory traced by that shape during gesticulation. We use eigenspace approach for compact, approximate representation of different hand shapes. Such a representation is robust as it is based on the general appearance of the hand, and is independent of existence of any specific feature. Such a representation is possible because of use of EigenTracker.

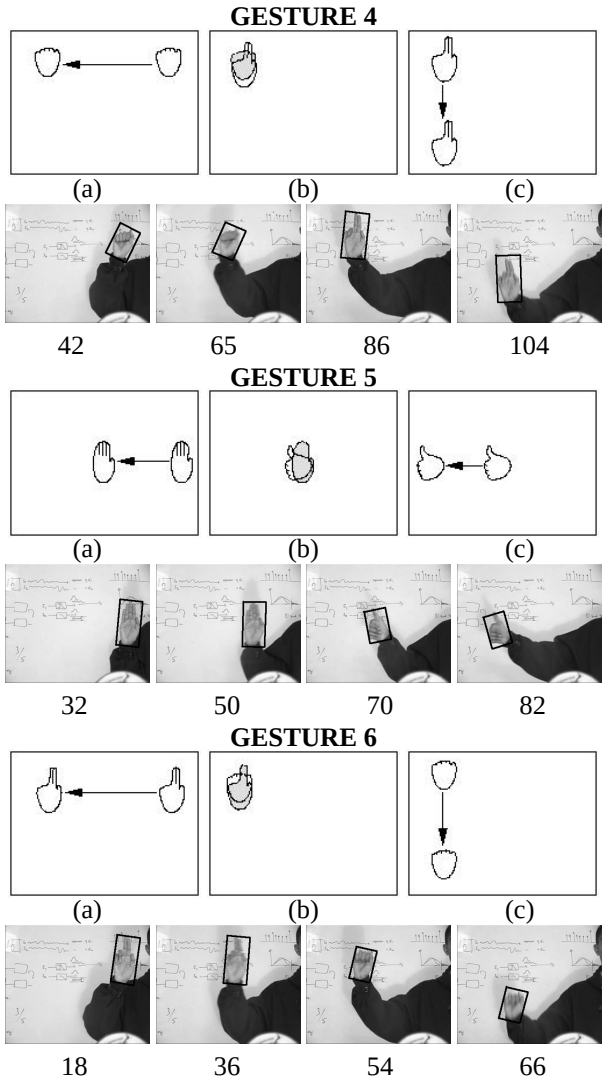


Figure 3. Gestures 4, 5, and 6 in the vocabulary.

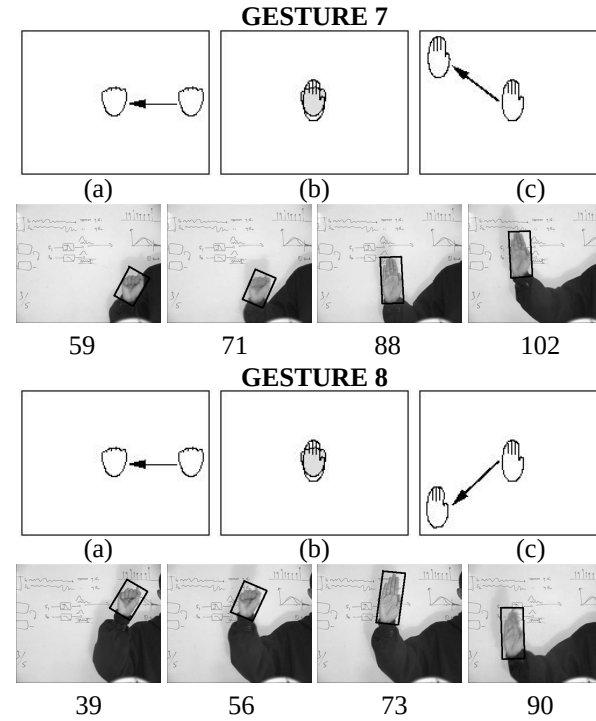


Figure 4. Gestures 7 and 8 in the vocabulary.

2.1. Modeling a Gesture

An EigenTracker gives as output a set of eigenspace reconstruction coefficients $\mathbf{c}$ and affine transformation coefficients $\mathbf{a}$. Depending upon the subsequent reconstruction error, the EigenTracker updates the eigenspace. Drastic change in the appearance of the gesticulating hand, caused by the change in the hand shape, results in large reconstruction error forcing an epoch change – constructing the appearance eigenspace afresh. An epoch change thus indicates a new shape of the gesticulating hand. The view of the hand at i^{th} epoch is stored as shape s_i. An eigenspace E_s is then constructed from properly scaled shapes s_i that are collected from different training gestures. The coefficients c_{si}, result of projecting the shape s_i on this eigenspace E_s, are used to represent the shapes in a unified manner.

After every epoch, the sequence of affine coefficients output by the EigenTracker gives the trajectory traced by that hand shape (s_i) in space. This trajectory t_i is modeled by a curve (line, spline or other higher order model) that is described by a set of parameters c_{ti}. Vector representation v_j of the gesture j, (*gesture vector* v_j) is obtained by stacking together these coefficient sets observed in the gesture, $v_j = [c_{si}\ c_{ti}]^T\ ,\ \forall i$, where i denotes the epoch number.

2.2. Choosing a Gesture Set

Gestures are classified according to the number of epoch changes required by the EigenTracker. Therefore, different classes of gestures differ in the size of the gesture vectors used to describe them. The gesture vocabulary is represented as $\bigcup_i \{k_i\}$, where k_i is the set of gestures that require i epoch-changes during the tracking phase. To recognise a query gesture v_t, the search is localised in that set k_i which has the same number of epoch-changes as in the query gesture.

Using the sample gestures available in the training set, the mean gesture vector (gesture template) $\overline{v}_j$ and corresponding correlation matrix Σ_j is calculated for every gesture j. To classify a query gesture represented by gesture vector v_t, we calculate its Mahalanobis distance d_{tj} from each of the mean gestures $\overline{v}_j$, given by,

$$d_{tj} = \left[(v_t - \overline{v}_j)^T \Sigma_j^{-1} (v_t - \overline{v}_j)\right]^{\frac{1}{2}}, \forall j \qquad (1)$$

The likelihood p_j of the given query gesture being classified as gesture j is given by

$$p_j = \frac{e^{-d_{tj}}}{\sum_k e^{-d_{tk}}} \quad (2)$$

In this scheme, the gesture-space is partitioned into smaller subspaces, each corresponding to a particular *class* of gestures. Each gesture is represented by a *region* in the gesture-subspace of corresponding gesture class, characterised by the mean gesture vector and the correlation matrix. In the gesture vocabulary, the gestures belonging to the same class should be chosen so that they occupy well separated regions in the corresponding gesture subspace. It is apparent that the minimum distance between two gestures belonging to the same class puts an upper bound on the accuracy of the recognition system. Higher the separation of the gestures in gesture-space, better is the performance of the recognition system. The following section describes the experimental results obtained using this scheme.

3. A Representative Gesture Set: Experiments

In this section, we present the results of our experimentation with a representative gesture set. Four different hand shapes are used to construct the gesture vocabulary. Figure 1(a) shows the four basic hand shapes - A, B, C, and D. These shapes were chosen since they significantly differ from each other in appearance, thus minimizing the possibility of incorrect shape identification. The gesture vocabulary consists of eight gestures. These gestures can be used to control application software such as Winamp©.

3.1. Gesture Set Modeling

Each gesture consists of two different hand shapes, requiring two epoch changes in the tracking phase. Figure 2 shows the first three gestures in the vocabulary. For every gesture, the upper row depicts the schematic, and the second row shows frames extracted from the actual video. Figures 3, 4 show gestures four to eight of the vocabulary, and follow the same convention as in Figure 2. Note that gesture pairs two-six, three-four, and seven-eight, involve identical hand shapes (in order) and differ only in the hand trajectories. Conversely, in gesture pairs one-five, two-three, and four-six, the hand traces identical trajectory but assumes different shapes. In spite of this apparent resemblance, the gestures in the vocabulary are well separated in *gesture-space*. Table 1 shows the distances between the gesture templates $\overline{v}_j, j = \{1, \cdots, 8\}$ of our gesture vocabulary.

3.2. Modeling a Gesture

For the gestures of our vocabulary, we use straight line approximation to the trajectory traced by a particular hand shape. In parametric form, the line l is represented as $a + r_1 x + r_2 y = 0$, where $\sqrt{{r_1}^2 + {r_2}^2} = 1$. Thus, $c_l = [a\ r_1\ r_2]^T$ is the set of parameters that completely describes the trajectory traced by the corresponding hand shape. We estimate the parameters a, r_1, r_2 using the total least squares minimization technique. Different hand shapes are represented using the set of parameters obtained by projecting them onto the eigenspace $\mathbf{E}_s$ (ref. sec. 2.1).

3.3. Training

To calculate the mean gesture vector and corresponding correlation matrix (for gesture recognition) eight test sequences were used for every gesture in the vocabulary. 64 training sequences were thus used in total. Since every gesture involves two hand shapes, a total of 128 hand shape images were collected from the training data. Figure 1(b) shows some of the shape images collected from the test data. (The black dots arise at points of faulty skin colour detection.) After using linear interpolation to normalise the size of each of the hand shape images to 100×100, singular value decomposition (SVD) was calculated for the entire set. It was observed that five most significant eigenvalues contributed more than 90% of the total energy, and contribution of each of the rest was marginal. This can be explained by the redundancy in the input hand shape images – four different hand shapes and 128 sample images. We therefore describe every hand shape by taking its projections on these five basis eigenvectors.

Each gesture is thus represented by a 16 element vector, $[\alpha_{11}\ \alpha_{12} \ldots\ \alpha_{15}\ \ \beta_{11} \ldots\ \beta_{13}\ \ \alpha_{21}\ \alpha_{22} \ldots\ \alpha_{25}\ \ \beta_{21} \ldots\ \beta_{23}]^T$, where α_{1i} and β_{1i} are parameters describing the hand shape and trajectory corresponding the first epoch, and α_{2i} and β_{2i} describe the shape and trajectory of the hand in the second epoch, respectively.

3.4. Gesture Recognition

Figures 5 and 6 show the intermediate steps in processing of gesture four from set seven using our scheme. Figure 5 shows a few frames of the tracker's output. The hand is marked with a tightly fitting bounding box. The tracker follows the hand with initial bounding box parameters and eigenspace till frame 85. At the end of processing frame 85, a large object reconstruction error forces an epoch change. A new bounding box is calculated in frame 86 ([4], [5]). The tracking commences with this new hand shape detected in frame 86. In Figure 6(a), on left, we show the shape of the hand - properly scaled – detected by the tracker in frame 42. Shown on the right, in Figure 6(a), is the linear approximation to the trajectory traced by this shape of the hand. On similar lines, Figure 6(b) shows the detected hand shape (after normalising the image size) and the linear approxi-

	GES. 1	GES. 2	GES. 3	GES. 4	GES. 5	GES. 6	GES. 7	GES. 8
GES. 1	**0**	1.5×10^8	5.8×10^7	6.7×10^7	8.5×10^7	1.1×10^8	7.8×10^7	1.7×10^8
GES. 2	3.3×10^8	**0**	1.3×10^7	5.4×10^7	1.9×10^9	2.3×10^7	1.6×10^6	3.8×10^8
GES. 3	3.0×10^8	2.0×10^7	**0**	6.7×10^7	1.9×10^9	3.5×10^7	3.2×10^7	6.3×10^8
GES. 4	3.8×10^8	4.8×10^7	6.1×10^7	**0**	2.0×10^9	8.0×10^6	7.6×10^7	5.7×10^8
GES. 5	4.1×10^7	1.8×10^8	7.5×10^7	7.2×10^7	**0**	1.2×10^8	8.6×10^7	5.8×10^8
GES. 6	4.0×10^8	3.9×10^7	6.1×10^7	8.6×10^6	2.0×10^9	**0**	8.7×10^7	3.4×10^8
GES. 7	7.3×10^7	8.1×10^7	2.2×10^7	1.2×10^8	7.6×10^8	1.5×10^8	**0**	1.1×10^8
GES. 8	5.1×10^7	1.1×10^8	1.1×10^8	4.2×10^7	5.6×10^8	6.8×10^7	1.5×10^8	**0**

Table 1. Mahalanobis distance between the template gestures

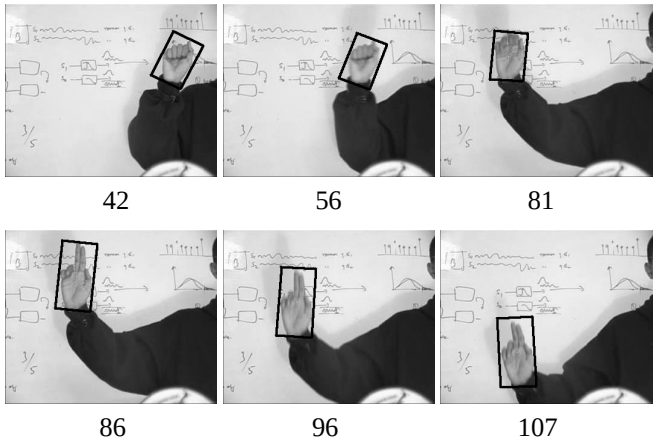
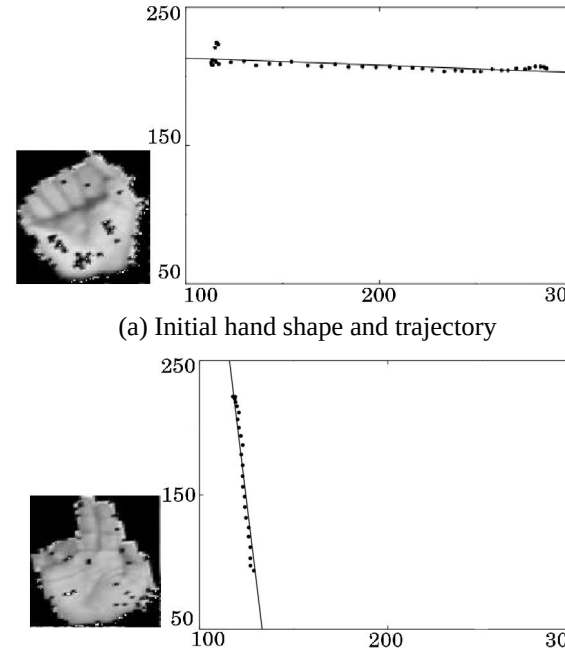


Figure 5. Output of predictive EigenTracker (gesture four, from set seven).



(a) Initial hand shape and trajectory

(b) Second hand shape and trajectory

Figure 6. Hand shapes and Linear approximation of trajectories (gesture four, from set seven).

mation to the trajectory in the second half of the gesture.

The performance of this framework for gesture recognition was tested using 64 gestures present in the training set, and 16 additional gestures which were not used during the training phase. Table 2 lists the Mahalanobis distances of gestures of set seven from template gestures. These gestures were used, among others, during the training phase to calculate the gesture templates. All these gestures are correctly recognised. Similar results were also observed for other gestures that were used during training. The Mahalanobis distances of gestures of set nine (which were not included in the training process) from the template gestures are listed in Table 3. As evident is this table, the system correctly recognises all these gestures. To conclude, the system recognised all the 80 gestures with 100% accuracy.

4. Conclusion

In this paper we present a novel approach to represent dynamic gestures by vectors. We also introduce the notion of *distance* between gestures that can facilitate proper choice of gestures in the vocabulary. We utilise the inter-gesture distance for gesture recognition. Further extensions of this work include applying this framework for two handed gesture recognition. Another interesting extension would be to best adapt the framework to an existing set of gestures.

References

[1] T. Ahmad, C. Taylor, A. Lanitis, and T. Cootes. Tracking and recognition of hand gesures using statistical shape models. *Image and Vision Computing*, 15:345 – 352, 1997.

[2] O. Al-Jarrah and A. Halawani. Recognition of gestures in arebic sign language using neuro-fuzzy sys-

	GES. 1	GES. 2	GES. 3	GES. 4	GES. 5	GES. 6	GES. 7	GES. 8
1	$\mathbf{0.79}$	1.5×10^8	1.2×10^8	7.9×10^7	3.8×10^7	1.2×10^8	8.6×10^7	1.5×10^8
2	2.5×10^8	$\mathbf{5.3}$	2.8×10^7	2.4×10^7	1.8×10^9	1.8×10^7	5.8×10^6	2.0×10^8
3	2.6×10^8	2.0×10^7	$\mathbf{1.3}$	3.8×10^7	2.0×10^9	3.7×10^7	2.9×10^6	6.2×10^7
4	3.8×10^8	8.3×10^7	6.2×10^7	$\mathbf{1.4}$	2.0×10^9	6.2×10^6	6.3×10^7	3.1×10^8
5	6.6×10^7	1.6×10^8	3.8×10^7	5.7×10^7	$\mathbf{0.97}$	1.1×10^8	8.2×10^7	1.2×10^8
6	4.4×10^8	7.3×10^7	4.7×10^7	1.2×10^7	2.1×10^9	$\mathbf{0.86}$	4.6×10^7	3.1×10^8
7	6.6×10^7	1.3×10^8	3.7×10^7	8.8×10^7	6.4×10^8	1.1×10^8	$\mathbf{0.51}$	2.7×10^8
8	4.0×10^7	1.1×10^8	1.1×10^8	3.4×10^7	6.7×10^8	5.6×10^7	1.6×10^8	$\mathbf{1.9}$

Table 2. Mahalanobis distance of gestures of set seven from template gestures

	GES. 1	GES. 2	GES. 3	GES. 4	GES. 5	GES. 6	GES. 7	GES. 8
1	$\mathbf{0.98}$	1.6×10^8	5.9×10^7	2.7×10^7	9.4×10^7	1.0×10^8	8.6×10^7	1.6×10^8
2	2.1×10^8	$\mathbf{0.26}$	2.1×10^7	4.4×10^7	1.6×10^9	2.9×10^7	1.5×10^7	2.1×10^8
3	3.0×10^8	5.1×10^7	$\mathbf{0.88}$	4.6×10^7	1.8×10^9	3.8×10^7	5.4×10^7	1.7×10^8
4	3.2×10^8	6.5×10^7	6.0×10^7	$\mathbf{0.77}$	1.8×10^9	5.4×10^6	6.5×10^7	1.6×10^8
5	5.9×10^7	1.8×10^8	7.4×10^7	6.9×10^7	$\mathbf{0.95}$	1.1×10^8	9.5×10^7	1.3×10^8
6	4.2×10^8	4.4×10^7	4.7×10^7	4.7×10^6	2.2×10^9	$\mathbf{0.99}$	7.1×10^7	2.9×10^8
7	3.6×10^7	8.9×10^7	4.5×10^7	1.0×10^8	6.8×10^8	1.3×10^8	$\mathbf{0.88}$	2.6×10^8
8	7.7×10^7	9.2×10^7	1.3×10^8	4.2×10^7	6.2×10^8	6.5×10^7	1.2×10^8	$\mathbf{0.97}$

Table 3. Mahalanobis distance of gestures of set nine (not used for training) from template gestures

tems. *Artificial Intelligence*, 133:117 – 138, 2001.

[3] M. J. Black and A. D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26(1):63 – 84, 1998.

[4] N. Gupta, P. Mittal, K. S. Patwardhan, S. Dutta Roy, S. Chaudhury, and S. Banerjee. Online Predictive Appearance-based Tracking. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 2004.

[5] N. Gupta, P. Mittal, K. S. Patwardhan, S. Dutta Roy, S. Chaudhury, and S. Banerjee. Condensation-based predictive eigentracking. *Pattern Recognition*, (Communicated).

[6] H. Hongo, M. Ohya, M. Yasumoto, and K. Yamamoto. Visual recognition of static/dynamic gesture: Face and hand gesture recognition for human-computer interaction. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 2921 – 2924, 2000.

[7] B. Min, H. Yoon, J. Soh, Y. Yang, and T. Ejima. Visual recognition of static/dynamic gesture: Gesture driven editing system. *Journal of Visual Languages and Computing*, 10:291 – 309, 1999.

[8] Y. Nam and K. Wohn. Recognition of hand gestures, with 3d, non-linear arm movement. *Pattern Recognition Letters*, 18:105 – 113, 1997.

[9] C. W. Ng and S. Ranganath. Real-time gesture recognition system and application. *Image and Vision Computing*, 20:993 – 1007, 2002.

[10] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677 – 695, July 1997.

[11] M. W. T. Kapuscinski. Hand gesture recognition for man-machine interface. In *Second workshop on Robot motion and control*, October 2001.

[12] J. Triesch and C. Malsburg. Classification of hand postures against complex backgrounds using elastic graph matching. *Image and Vision Computing*, 20:937 – 943, 2002.

[13] M. Yeasin and S. Chaudhuri. Visual understanding of dynamic hand gestures. *Pattern Recognition*, 33:1805 – 1817, 2000.

[14] Y. Zhua, G. Xu, and D. Kriegman. A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction. *Computer Vision and Image Understanding*, 85:189 – 208, 2002.