Accurate Recognition of Large Number of Hand Gestures

Atid Shamaie **Alistair Sutherland**
Machine Vision Group, Centre for Digital Video Processing
School of Computer Applications, Dublin City University,
Dublin 9, Ireland
{ashamaie, alistair}@computing.dcu.ie

Abstract: A hierarchical gesture recognition algorithm is introduced to recognise a large number of gestures. Three stages of the proposed algorithm are based on a new hand tracking technique to recognise the actual beginning of a gesture using a Kalman filtering process, hidden Markov models and graph matching. Processing time is important in working with large databases. Therefore, special cares are taken to deal with the large number of gestures, which are partially similar.

Keywords: HMM, PCA, Graph Matching, Kalman Filter, Tracking.

1. Introduction

The use of hand gesture as a more natural way in Human Computer Interaction has been addressed in the literature [1]. Statistical methods are from the very popular approaches to the recognition of hand gestures. However, many approaches have been introduced. Spatio-temporal hand gesture recognition using neural networks [2] [3], temporal models for gesture recognition [4], spatial modelling of gestures [5] [6], recognition of gestures using hidden Markov models (HMM) [5] [7] [8], parametric hidden Markov models [9], HMM-based threshold models for gesture recognition [10], Principal Component Analysis [11] [12] [13] [14], position-based gesture recognition [15], tracking interacting hands using the Bayesian networks [16] and many other techniques have been used to deal with the problem of gesture recognition. Also the graphs [17] and graph matching [18] [19] techniques are used in computer vision [20] and gesture recognition [21].

In this paper we will discuss the problem of gesture recognition in the case of large number of gestures and a new hierarchical algorithm which is based on a dynamic model of hand movements, hidden Markov models and Graph matching is introduced.
In the next section the main problem is addressed. In Section 3 some special considerations regarding the number of gestures are discussed. Section 4 and the subsections are dedicated to our new hierarchical algorithm. In Section 5 some experimental results are presented. And finally some discussion and conclusions are stated at the end of the paper.

2. Problem statement

A sequence of images containing a hand gesture is at our main focus. By using the colour segmentation one can extract the hand from the background in an image.
Herein, it is assumed that in the sequence of input images to the algorithm the hand has been segmented from the background. This leads to the following definition of the problem:
" Given a sequence of images containing a hand gesture, find a gesture in a large database of predefined gestures which is the most similar to that."

One approach for working with high dimensional data uses Principal Component Analysis (PCA) [21]. We use very low-resolution images (32x32 pixels) and every frame of the input sequence is mapped onto a point in a 1024 dimensional space. A sequence of images is mapped onto a sequence of points, which form a trajectory. Principal Component Analysis for reducing the dimensionality, vector quantization for code- word extraction, hidden Markov models [22] for temporal analysis of image sequences, and Graph Matching for final decision making are the methods used in our algorithm.

3. Special considerations

Generally, since we have a large database of predefined gestures the similarity between the gestures is high. Therefore, a small amount of noise may change the result of recognition. Particularly, we have about 80 different recognisable shapes for a hand and every gesture should contain a combination of these shapes. Therefore, different gestures may be alike partially.

The other important point is the variations in the gestures. Every sample of a gesture varies from the other samples and we have to use a technique that is able to deal with variations. However, the whole algorithm should be able to deal with the variations and the similarities together.

Also, working with a large database of gestures involve extensive computations and finding the most similar gesture to a given unknown gesture by comparing all the gestures takes a long processing time.

We deal with these problems by a hierarchical algorithm. In the proposed algorithm we use two levels (stages) of filtering and a third level for decision making. The filtering levels are used to extract the most probable gestures from the database and the final level's decision making is based on a selection between a few gestures passed through the previous filters.
In the first level, a filtering technique based on the recognition of the beginning shape of a gesture and a dynamic model is used to select about *1/g* of the gestures of the database where *g* may vary to find a trade-off between the running time and the recognition rate. This reduces the processing time very much.

In the second level of filtering, by using the HMMs we deal with variations and choose a small set of gestures to pass to the third level. And in the third level the similarity of the passed gestures is considered and dealt with by using the Graph Matching technique [21].

4. The new hierarchical algorithm

In this algorithm we have a hierarchical recognition process which uses a multilevel trained model. The first levels of the training and recognition phase depend on the beginning shape of the hand. By the beginning shape we mean the actual hand shape that a gesture starts with. Recognising the beginning shape when the hand starts from a rest position is a problem. A technique is introduced for this recognition.

4.1. Hand movement tracking

We assume a hand gesture starts from a neutral position. This condition can be the hand hanging by the side or the hand on lap. We consider this as the hand out of region of interest (ROI). The region of interest is defined as the area in which the hand gesture must be done to be recognised by the system. When the person starts to do a hand gesture he/she moves the hand up and starts to do the gesture.

Before the gesture starts the hand should move into the ROI. Every gesture starts with a particular shape of hand. So, normally, from the neutral position to the start point of the

gesture the position and the shape of hand is changed toward the beginning position and shape of the gesture.

At a moment when it is approved, by the brain, that the hand has reached to the correct position and shape to start the gesture the hand has a very short stop (a small fraction of a second) and then it moves toward the end of the gesture. At the end of gesture again the hand may have a short stop and then moves out of the ROI. However, this is not true in all the cases. This is not important for us, because whatever the gesture is, the system is trained for the gesture from the beginning to the end. If the end of a gesture is just moving out of the ROI it is the same for the training and the recognition phases.

A tracking algorithm is introduced, which detects the moment that hand has a short stop. We use a model in which the position, velocity and acceleration of the hand are modelled by a Kalman filtering process [23].

Let $x_{(t)}$ denote the trajectory of hand movement in a two-dimensional image plane where t is the time variable. This function is discretised by sampling with $f = \frac{1}{h}, \quad h > 0$ sampling rate. Therefore, $x_k = x_{(kh)} \quad k = 0,1,\ldots$. $x_{(t)}$ can be assumed to have continuous first and second order derivatives. For small values of h the position and velocity vectors are calculated by

$$x_{k+1} = x_k + h\dot{x}_k + \frac{1}{2}h^2\ddot{x}_k \tag{1}$$

$$\dot{x}_{k+1} = \dot{x}_k + h\ddot{x}_k \tag{2}$$

where $\dot{x}_k = \dot{x}_{(kh)} \quad k = 0,1,\ldots$ is the velocity, the first derivative,

and $\ddot{x}_k = \ddot{x}_{(kh)} \quad k = 0,1,\ldots$ is the acceleration, the second derivative.

We define the position of hand by the centre of the rectangle containing the hand, Figure 1.

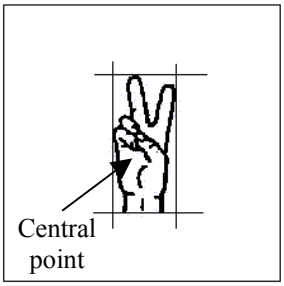

Figure 1. Position of hand is defined by the centre of the rectangle.

Therefore, the position is denoted by the vector

$$x_k = \begin{bmatrix} x_k^h \\ x_k^v \end{bmatrix}$$

where x_k^h is the horizontal coordinate of the hand centre, and x_k^v is the vertical coordinate of the hand centre. However, we can only observe the position of hand in the image frame and the other parameters, velocity and accelerations are not observable. Therefore, the matrix H is defined as

$$H = \begin{bmatrix} I & 0 & 0 \end{bmatrix}, \quad z_k = H x_k . \tag{3}$$

The hand tracking model takes on the following linear stochastic description

$$\begin{cases} x_{k+1} = \Phi x_k + w_k \\ z_k = H x_k + v_k \end{cases} \tag{4}$$

where

$$x_k = \begin{bmatrix} x_k^h \\ \dot{x}_k^h \\ \ddot{x}_k^h \\ x_k^v \\ \dot{x}_k^v \\ \ddot{x}_k^v \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 & & & \\ 0 & 1 & h & & & \\ 0 & 0 & 1 & & & \\ & & & 1 & h & \frac{1}{2}h^2 \\ & & & 0 & 1 & h \\ & & & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

A detection parameter is defined based on the velocity of the hand. When the hand stops the velocity reaches zero. In fact, a well-chosen small threshold gives appropriate detection accuracy. In a two-dimensional image frame

$$v_d^2 = v_h^2 + v_v^2 \tag{5}$$

where

$v_h = \dot{x}_k^h$: horizontal velocity,

$v_v = \dot{x}_k^v$: vertical velocity.

For a small chosen $\varepsilon > 0$ if $v_d < \varepsilon$ we conclude that the hand has stopped. From the stop point the system records the hand gesture to be recognised.

4.2. Training phase

In the first level of the training phase, a common seven-dimensional eigenspace is formed by using the beginning frames of all the gestures in the training set. This is done by using PCA. In the eigenspace every image is mapped onto a point and the sets of similar shapes make sets of clusters in the subspace, Figure 2. The mean point of each cluster is considered as the representative of the cluster in this space.

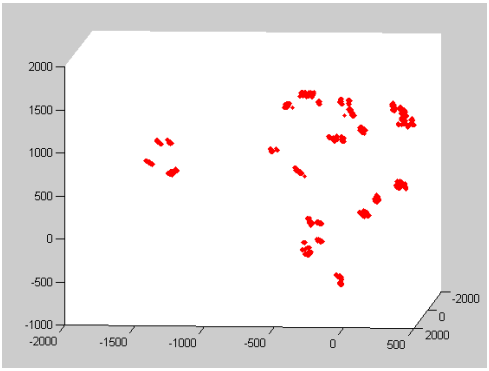

Figure 2. A 3 dimensional common space formed by 26 different shapes of hand.

In the second level of the training phase another common space is formed by the full image sequence of all the gestures in the training set. Using PCA this subspace is made and

the projection of the image sequences in this space form the trajectories (manifolds) of the different gestures.

Now we have to allocate codevectors to the groups of points. Vector quantization can extract the codevectors. Based on the number of the gestures in the database the number of codevectors can be chosen so that all the gestures are exclusively recognisable by unique sequences of the codevectors. But, since the vector quantization is an extremely time consuming process, in the case of large number of data points and codevectors, one can extract the codevectors for the manifold of each gesture in the common space separately as opposed to treating the whole data at once. At the end by combining all the codevectors of all the manifolds and allocating a unique symbol to each, a large alphabet of codevectors is formed and all the gestures are uniquely recognisable by a sequence of symbols. However, since different gestures are similar in some parts and we have extracted the codevectors in the common space separately, one can conclude that many codevectors are very close together and can be represented by just one codevector. This is true and from the processing time point of view the importance of reducing the number of codevectors is more obvious.

We can use a second stage vector quantization to reduce the number of codevectors. However, it has negative effect on the recognition rate. So, there should be a trade-off between the processing time and the desired recognition rate. In the section of experimental results this is explained on a real example. Figure 3 shows the manifolds and the extracted codevectors for three gestures.

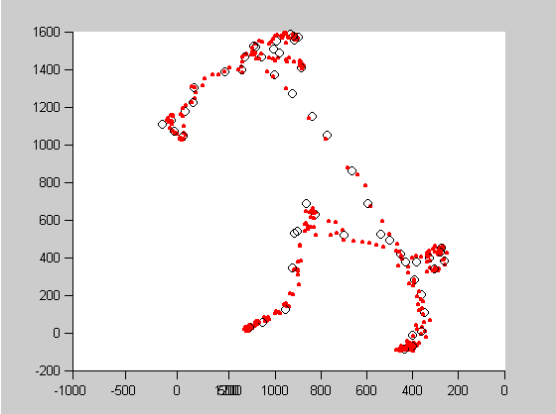

Figure 3. An illustration of 3 manifolds and the extracted codevectors (small circles).

A left-to-right hidden Markov model (HMM), Figure 4, for every gesture is trained by using the sequences of symbols constituting the manifold of the gesture in the eigenspace. For all the samples of a gesture the codevectors are extracted and the sequences of the associated symbols are used to train the HMM. So, for every gesture there is a trained HMM at the end.

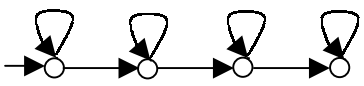

Figure 4. A left-to-right Hidden Markov Model used in the algorithm.

Because of high similarity between the gestures we have to consider separate eigenspaces for each gesture in order to distinguish the gestures as much as possible. For this purpose, by using all the samples of a gesture in the training set an individual eigenspace is made for every gesture. The projection of the gestures into their own subspaces form the Main manifold of every subspace. Then by using the vector quantization a set of codevectors are

extracted for each manifold, Figure 5. Every manifold can be represented as a graph whose vertices are the extracted codevectors.

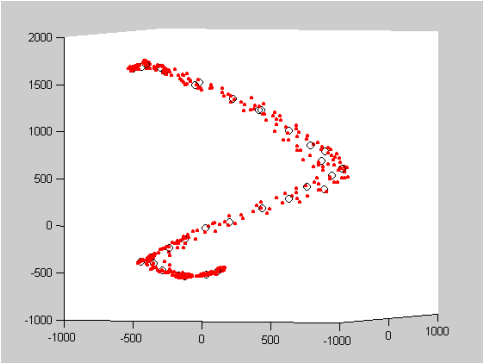

Figure 5. An illustration of a Main manifold in its own subspace and the extracted codevectors (small circles).

4.3. Recognition phase

In order to recognise an unknown gesture a hierarchical algorithm is used. At each level a group of gestures are selected to be passed to the next level. This helps to improve the speed of recognition greatly. Figure 6 is an illustration of the hierarchy of selections.

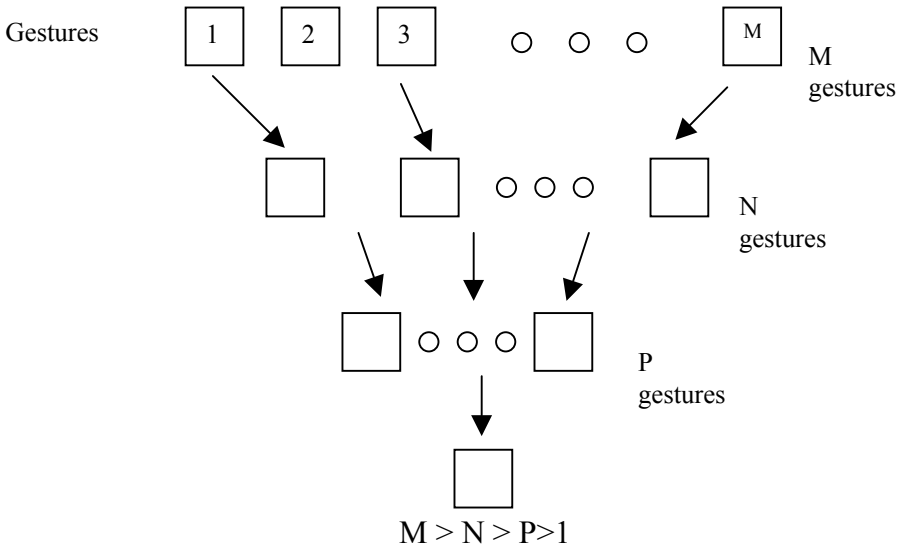

Figure 6. Hierarchy of selections in the recognition algorithm.

Level 1

At this level the beginning image of a gesture is projected into the eigenspace made at the first level of the training phase, which maps onto a point. By finding the Euclidean distance of this point to all the representatives in this space a list of the representatives is formed, which is sorted in ascending order based on the Euclidean distances. The nearest representative at the top of the list represents a group of gestures that start with the same shape as the unknown gesture. However, because of variation in the position, the angle of hand and, of course, because of noise, this is not the best estimate and one should consider more than one representative. By taking α representatives from the top of the list a group of gestures starting with one of the selected shapes is passed to the next level. Since many gestures may

start with the same shape of the hand, normally, the number of gestures passed to the second level is larger than α.

Level 2

By projecting the input gesture into the second common eigenspace formed in the second level of the training phase the nearest sequence of symbols is extracted. The trained HMMs of the gestures forwarded from the first level are employed to calculate the likelihood of the extracted sequence. One can consider the HMM that results in the largest likelihood as the best match. However, because of the similarity of the gestures many gestures may have the same sequence of extracted codevectors (symbols) in some parts. Also, the large number of gestures makes the extracted sequence of codevectors very similar and a small amount of noise can change the extracted sequence of codevectors.

Therefore, the gestures are sorted based on their likelihood at this stage. The correct gesture has either the highest likelihood or a small deviation from the highest one. So, a well-chosen margin gives a few gestures with a small deviation from the greatest likelihood. These gestures are chosen to be compared carefully with the unknown gesture.

Level 3

By projecting the unknown gesture into the eigenspaces of the selected gestures we try to find the best match. The projections of the gesture in the subspaces form the individual manifolds. As in [21] each manifold is estimated by a graph. Therefore, there are two graphs in every subspace. The main graph, say G_i, that represents the main manifold of the subspace and the graph of the manifold of the unknown gesture, G_i' in the subspace *i*. The best match of every pair of the graphs $\Phi_i = Match(G_i, G_i')$ is determined [21]. The following likelihood is used to find the best match,

$$L_i = \frac{n_i}{2N(m_i + 1)} \tag{6}$$

where *N* is the number of vertices of every predefined graph (usually the predefined graphs have the same number of vertices,) n_i denotes the number of vertices of the bipartite subgraph after the matching process, and m_i is the mean of the distances of the connected vertices in the final subgraph. The highest likelihood represents the best match.

5. Experimental results

100 different gestures were created by a combination of about 35 shapes mostly selected from the American Sign Language. The gestures start from a shape and end in another shape. In Figure 7, a few gestures is shown. For every gesture, 10 samples were captured. Half of the samples (500) were used for training and the rest for recognition. In the first level of the training phase the beginning shapes of the training gestures were used which are almost similar to the shapes defined in American Sign Language. Therefore, 26 clusters are formed in the first common eigenspace. In the second level of the training phase 500 out of 1000 samples were used to train 100 HMMs. A common eigenspace of all the training samples were formed and by projecting the training samples into this space, Figure 8, and using vector quantization 3200 codevectors were extracted. A question here is how much variation in the samples of a gesture exists.

Although, not deliberately did we vary the samples of each gesture, the graphs of the samples show significant variations, Figure 9. Therefore, this variation makes some

differences in the sequence of alphabets extracted in the second stage. And since the number of gestures is large and the gestures are similar in some parts, this variation may cause serious trouble and totally change the result of recognition. For this reason we used the HMMs in the second level. It is able to deal with the variations very well [22] and the similarity of the gestures is considered in the third level. But the problem is that the extraction of the codevectors for a gesture in a common space with a large number of codevectors is time consuming. We will explain this problem shortly.

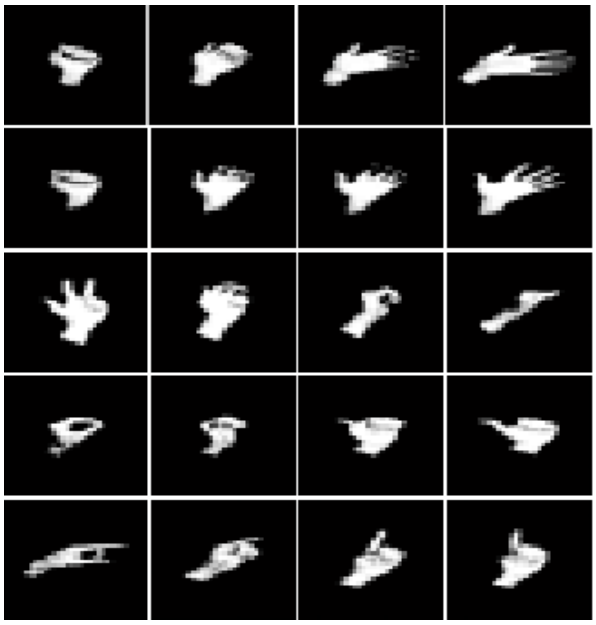

Figure 7. Sample frames of some of the gestures used in the experiments.

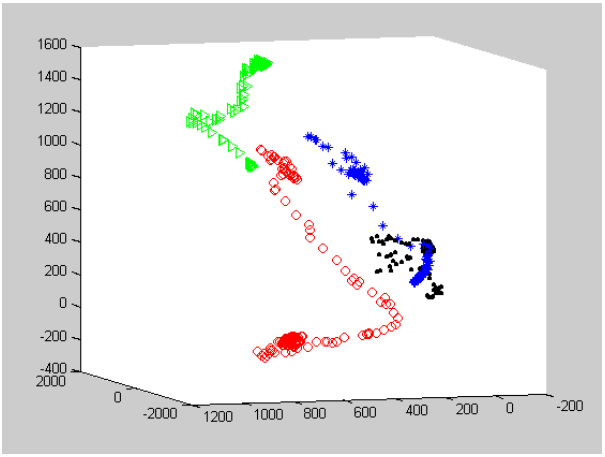

Figure 8. The projection of some gestures into the common space.

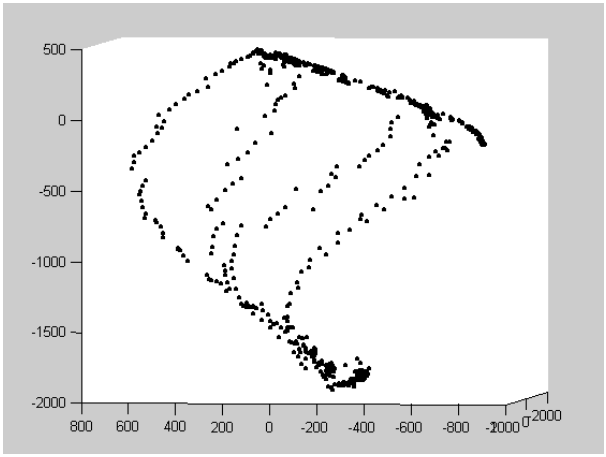

Figure 9. The variations of different samples of a gesture.

By using the training samples of each gesture an individual eigenspace was formed for every gesture. These eigenspaces are used in the third level of the recognition phase. At this level we are trying to make decision between a few predefined gestures. So, by considering the individual eigenspaces the similarities and the differences between the gestures are more expressed than in the case of a common space. Indeed, the graph matching algorithm is fast in the case of a small number of graphs, and also it is able to deal with the similarities in the gestures.

The other 500 samples of the 100 created gestures were used in our experiments and the recognition rate of 89.6% obtained. In this experiment, in the second stage of training, we extracted 32 codevectors for all the samples of each gesture. At the end we combined the 3200 codevectors and used them in the second level of the recognition phase with HMMs.

This causes a time consuming process to extract the sequence of codevectors for a given gesture. In a second experiment, by applying the second level of Vector Quantization to reduce the number of codevectors in the common space we got 1024 codevectors. The processing time reduced to almost one third of the first experiment but the recognition rate falls to %88.6. However, this is just one percent less recognition, while the speed of processing has increased dramatically.

Now the question is that what happens if the number of gestures increases. In this example if the number of gestures increases to 1000, more codevectors are needed in the common eigenspace of the second level. As it has been shown that HMM works well in a large number of classes of data [22] the important problem is the running time to extract the codevectors for a given gesture in the second stage of recognition.

Also, because of occlusion, if there is some missing points in the graph of gestures that causes missing codevectors in the extracted sequence, what happens to the recognition process. These problems will be considered in our next papers.

Note

In all the processes seven-dimensional eigenspaces were considered based on the results in [21].

References

[1] R. Cipolla and A. Pentland, *Computer Vision for Human-Machine Interaction,* Cambridge University Press, 1998.

[2] M. Su, H. Huang, C. Lin, C. Huang, and C. Lin "Application of Neural Networks in Spatio-Temporal Hand Gesture Recognition," Proc. of the IEEE World Congress on Computational Intelligence, USA, 1998.

[3] D. Lin, "Spatio-Temporal Hand Gesture Recognition Using Neural Networks," Proc. of the IEEE World Congress on Computational Intelligence, USA, 1998.

[4] R. Bowden and M. Sarhadi, "Building Temporal Models for Gesture Recognition," Proc. of the British Machine Vision Conference 2000, Vol 1, University of Bristol, UK, September 2000.

[5] V. Pavlovic, R. Sharma, and T. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," IEEE Trans. Patt. Anal. Mach. Intell., Vol. 19, No. 7, July 1997.

[6] J. Davis and M. Shah, "Toward 3-D Gesture Recognition," Int'l Journal of Pattern Recognition and Artificial Intelligence, Vol. 13, No. 3, May 1999.

[7] T. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," Proc. of the Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995.

[8] J. Schlenzig, E. Hunter, and R. Jain, "Recursive Identification of Gesture Inputs Using Hidden Markov Models," Proc. of the Second IEEE Workshop on Applications of Computer Vision, Sarasota, Dec. 5-7, 1994.

[9] A. D. Wilson and A. Bobick, "Parametric hidden Markov models for Gesture Recognition," IEEE Trans. Patt. Anal. Mach. Intell., Vol 21, No 9, September 1999.

[10] H. Lee and J. Kim, " An HMM-Based Threshold Model Approach for Gesture Recognition," IEEE Trans. Patt. Anal. Mach. Intel., Vol 21, No 10, October 1999.

[11] M. V. Lamar, M. S. Bhuiyan, and A. Iwata, "Hand Gesture Recognition Analysis and An Improved CombNET-II," Proc. of the IEEE Int'l Conf. Systems, Man and Cybernetics, Vol. 4, Tokyo,1999.

[12] B. Moghaddam, W. Wahid, and A. Pentland, "Beyond Eigenfaces: Probabilistic Matching for Face Recognition," Proc. of the 3rd IEEE Int'l Conf. Automatic Face & Gesture Recognition, Nara, Japan, April 1998.
[13] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Detection," Proc. of the 5th Int'l Conf. Computer Vision, Cambridge, MA, June 1995.
[14] B. Moghaddam, "Principal Manifolds and Bayesian Subspaces for Visual Recognition," Proc. of the 7th IEEE International Conf. Computer Vision, ICCV'99, September, 1999.
[15] C. W. Ng and S. Ranganath, "Gesture Recognition via Pose Classification," Proc. of the Int'l Conf. Pattern Recognition ICPR'00, Barcelona, Spain, September 2000.
[16] J. Sherrah and S. Gong "Resolving Visual Uncertainty and Occlusion through Probabilistic Reasoning," Proc. of the British Machine Vision Conference, Vol. 1, University of Bristol, UK, September 2000.
[17] R. Diestel, *Graph Theory*, Springer-Verlag New York Inc., 1997.
[18] M. Karpinski and W. Rytter, *Fast Parallel Algorithms for Graph Matching Problems,* Oxford University Press, Oxford, 1998.
[19] H. Bunke and K. Shearer, "A Graph distance metric based on the Maximal Common Subgraph," Pattern Recognition Letters, Vol. 19, 1998.
[20] H. Bunke, "Recent Developments in Graph Matching," Proc. of the Int'l Conf. Pattern Recognition ICPR'00, Barcelona, Spain, Sept. 2000.
[21] A. Shamaie and A. Sutherland, "Graph-Based Matching of Occluded Hand Gestures," Proc. of the Applied Imagery Pattern Recognition 2001 (AIPR'01), Washington, DC, October 2001.
[22] F. Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, 1997.
[23] C. K. Chui and G. Chen, *Kalman Filtering*, Third Edition, Springer-Verlag, Berlin, 1999.