FACE RECOGNITION USING AN EMBEDDED HMM

Ara V. Nefian and Monson H. Hayes III

Georgia Tech Lorraine
2-3 rue Marconi, Metz, France 57070
{ara, mhh3}@eedsp.gatech.edu

ABSTRACT

Hidden Markov Models (HMM) have been successfully used for speech and action recognition where the data that is to be modeled is one-dimensional. Although attempts to use these one-dimensional HMMs for face recognition have been moderately successful, images are two-dimensional (2-D). Since 2-D HMM's are too complex for real-time face recognition, in this paper we present a new approach for face recognition using an embedded HMM and compare this new approach to the eigenface method for face recognition, and to other HMM-based methods. Specifically, an embedded HMM has equal or better performance than previous methods, with reduced computational complexity.

1. INTRODUCTION

Face recognition from still images and video sequences is emerging as an active research area with numerous commercial and law enforcement applications. Face recognition systems can be used to allow access to an ATM machine or a computer, to control the entry of people into restricted areas, and to recognize people in specific areas (banks, stores), or in a specific database (police records). A robust face recognition system must operate under a variety of conditions, such as varying illuminations and backgrounds, and it must be able to handle non-frontal facial images of males and females of different ages and races.

Previous approaches to face recognition [1] include geometric feature-based methods, template-based methods [2], [3] [4], and more recently model-based methods [5], [6]. The most significant facial features of a frontal face image include the hair, forehead, eyes, nose and mouth. Furthermore, these features occur in a natural order, from top to bottom, even if the images undergo small rotations in the image plane, and/or rotations in the plane perpendicular to the image plane. Therefore, the image of a face may be modeled using a one-dimensional HMM by assigning each of these regions to a state as illustrated in Figure 1. In this model, the states themselves are not directly observable. What is observed are observation vectors that are statistically dependent upon the state of the HMM. These vectors are obtained from L rows that are extracted sequentially from the top of the image to the bottom. Since the length of each row is fixed, and the height of a face image is proportional to its width, this HMM is restricted to fixed-size face images. Although used to model two-dimensional data, this one-dimensional HMM achieved recognition rates of about 85% [5] [7] [8].

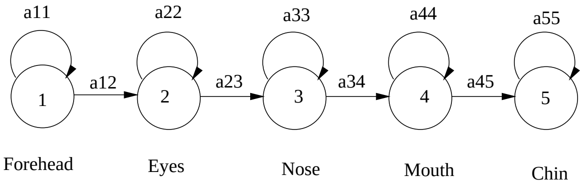

Figure 1: HMM for face recognition

This one-dimensional model was extended by Samaria to a structure that he referred to as a pseudo 2-D HMM by adding a *marker block* at the end of each line in the image, and introducing an additional *end-of-line state* at the end of each horizontal HMM as shown in Figure 2 [9]. The end-of-line states were allowed two possible transitions: one back to the beginning of the same row of states, and one that transitions to the next row of states. By setting the initial standard deviation of the end-of-line states to be close to zero, and the means close to the intensity of the end-of-line marker block, the state topology was preserved, and the parameters of the end-of-line states were unaltered after re-estimation. Samaria also considered a pseudo 2D-HMM that involved removing end-of-marker blocks as shown in Figure 3. Unlike the previous model, this topology allows transitions to a new super state from a frame that is not at the end of a row and, consequently, does not preserve the two-dimensional structure of the data. However, Samaria reported similar recognition results for both models, which were between 90 and 95 percent.

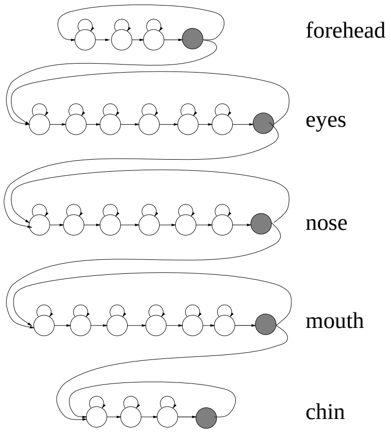

Figure 2: A one-dimensional HMM with end-of-line states.

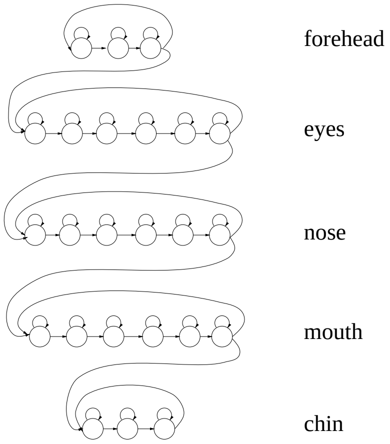

Figure 3: One-dimensional HMM without end-of-line states.

In this paper, we describe a new approach to face recognition using an embedded HMM as introduced by Kuo and Agazzi for character recognition [10]. Unlike previous HMM approaches to face recognition, which use pixel intensities to form the observation vectors, our embedded HMM uses observation vectors that are composed of two-dimensional Discrete Cosine Transform (2D-DCT) coefficients. Compared to template-based methods, and one-dimensional HMMs, our proposed system offers a more flexible framework for face recognition, and can be used more efficiently in scale invariant systems.

2. AN EMBEDDED HMM

A one-dimensional HMM is a Markov chain with a finite number of unobservable states [11]. Although the Markov states are not directly observable, each state has a probability distribution associated with the set of possible observations. Thus, when the HMM is in state i, the output (observation) is determined according to a given conditional probability density function, often Gaussian or a Gaussian mixture. What is necessary to statistically characterize an HMM is a state transition probability matrix, an initial state probability distribution, and a set of probability density functions associated with the observations for each state.

A one-dimensional HMM may be generalized, to give it the appearance of a two-dimensional structure, by allowing each state in a one-dimensional HMM to be an HMM. In this way, the HMM consists of a set of *super states*, along with a set of *embedded* states. The super states may then be used to model two-dimensional data along one direction, with the embedded HMM modeling the data along the other direction. This model differs from a true two-dimensional HMM since transitions between the states in different super states are not allowed. Therefore, this is referred to as an embedded HMM. The elements of an embedded HMM are:

- The number of super states, N_0, and the set of super states, $\mathbf{S}_0 = \{S_{0,i}\}\ 1 \leq i \leq N_0$.
- The initial super state distribution, $\mathbf{\Pi_0} = \{\pi_{0,i}\}$, where $\pi_{0,i}$ are the probabilities of being in super state i at time zero.
- The super state transition probability matrix,

  $$\mathbf{A_0} = \{a_{0,ij}\}$$

  where $a_{0,ij}$ is the probability of transitioning from super state i to super state i.
- The parameters of the embedded HMMs, which include
  - The number of embedded states in the kth super state, $N_1^{(k)}$, and the set of embedded states, $\mathbf{S}_1^{(k)} = \{S_{1,i}^{(k)}\}$.
  - The initial state distribution, $\mathbf{\Pi}_1^{(k)} = \{\pi_{1,i}^{(k)}\}$, where $\pi_{1,i}^{(k)}$ are the probabilities of being in state i of super state k at time zero.
  - The state transition probability matrix,

    $$\mathbf{A}_1^{(k)} = \{a_{1,jk}^{(k)}\}$$

    that specifies the probability of transitioning from state k to state j.

- Finally, there is the state probability matrix,

$$\mathbf{B}^{(k)} = \{b_i^{(k)}(\mathbf{O}_{t_0,t_1})\}$$

for the set of observations where $\mathbf{O}_{t_0,t_1}$ represent the observation vector at row t_0 and column t_1. In a *continuous density* HMM, the states are characterized by continuous observation density functions. The probability density function that is typically used is a finite mixture of the form

$$b_i^{(k)}(\mathbf{O}_{t_0,t_1}) = \sum_{m=1}^{M} c_{im}^{(k)} N(\mathbf{O}_{t_0,t_1}, \mu_{im}^{(k)}, \mathbf{U}_{im}^{(k)}) \quad (1)$$

where $1 \leq i \leq N_1^{(k)}$, $c_{im}^{(k)}$ is the mixture coefficient for the mth mixture in state i of super state k. $N(\mathbf{O}_{t_0,t_1}, \mu_{im}^{(k)}, \mathbf{U}_{im}^{(k)})$ is a Gaussian pdf with mean vector $\mu_{im}^{(k)}$ and covariance matrix $\mathbf{U}_{im}^{(k)}$.

Let $\mathbf{\Lambda}^{(k)} = \{\mathbf{\Pi}_1^{(k)}, \mathbf{A}_1^{(k)}, \mathbf{B}^{(k)}\}$ be the set of parameters that define the k^{th} super state. Using a shorthand notation, an embedded HMM is defined as the triplet

$$\lambda = (\mathbf{\Pi_0}, \mathbf{A_0}, \mathbf{\Lambda}). \quad (2)$$

where $\mathbf{\Lambda} = \{\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \ldots, \mathbf{\Lambda}^{(N_0)}\}$. Although more complex than a one-dimensional HMM, an embedded HMM is more appropriate for data that is two-dimensional, and has a complexity proportional to the sum of the squares of the number of states, $\sum_{k=1}^{N_0}(N_1^{(k)})^2$. The state structure of the face model and the non-zero transition probabilities of the embedded HMM are shown in Figure 4. Each state in the overall top-to-bottom HMM is assigned to a left-to-right HMM. This model is appropriate for face images since it exploits an important facial characteristic: frontal faces preserve the same structure of "super states" from top to bottom, and also the same left-to right structure of "states" inside each of these "super states". Compared with the other structures, an embedded HMM has the following advantages:

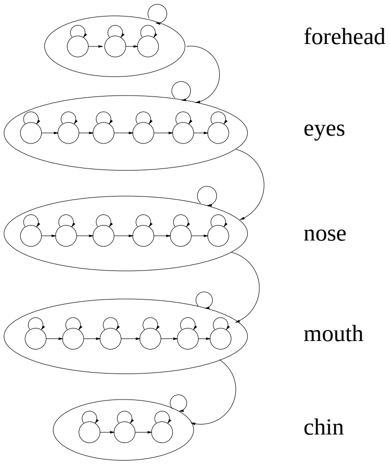

Figure 4: Embedded HMM for face recognition

1. The complexity is reduced both in terms of training and recognition,
2. Better initial estimates of the model parameters that can be obtained,
3. The two-dimensional structure of the data is naturally preserved without using extra frames or end-of-line states that increase the complexity of the model.

The states of the embedded HMM are described by single density Gaussian pdf

$$b_i^{(k)}(\mathbf{O}_{t_0,t_1}) = N(\mathbf{O}_{t_0,t_1}, \mu_i^{(k)}, \mathbf{U}_i^{(k)}) \quad (3)$$

and the covariance matrix is assumed to be diagonal.

3. THE OBSERVATION VECTORS

The observation sequence is generated using the technique shown in Figure 5, where a $P \times L$ window scans the image left to right, and top to bottom. The overlap between adjacent windows is M lines in the vertical direction and Q columns in the horizontal direction.

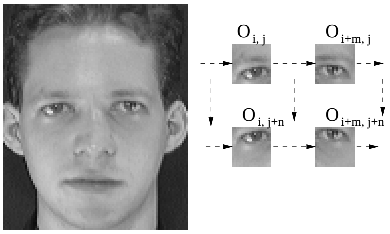

Figure 5: Face image parameterization and blocks extraction

In [9], the observation vectors consist of all the pixel values from each of the blocks, and therefore the dimension of the observation vector is $L \times P$. The use of the pixel values as observation vectors has two important disadvantages. First, pixel values do not represent robust features since they tend to be sensitive to image noise as well as image rotations or shifts, and changes in illumination. Second, the large dimension of the observation vector leads to high computational complexity of the training and recognition stages. This can be critical for a face recognition system that operates on a large database, or when the recognition system is used for real time applications.

In our embedded HMM, the observation vectors were formed from the 2D-DCT coefficients of each image block. The compression and decorrelation properties of the 2D-DCT for natural images makes it suitable for their use as observation vectors. Specifically, the coefficients within a rectangular window over the lowest frequencies in the 2D-DCT domain, where most of the image energy is found, were used as observation vectors. Using 2D-DCT coefficients instead of pixel values reduces dramatically the size of the observation vectors and, therefore, decreases the complexity of the recognition system. For our experiments, $L = 8$, $P = 10$ and six 2D-DCT coefficients from each block are used as observation vectors. Therefore, the size of the observation vectors is reduced over 13 times compared to the method that uses the pixel intensities as the observation vector.

4. TRAINING THE FACE MODELS

Each individual in the database is represented by an embedded HMM. A set of images representing different instances of the same face were used for training. The observation vectors extracted from each block were used to train the models as follows (Figure 6):

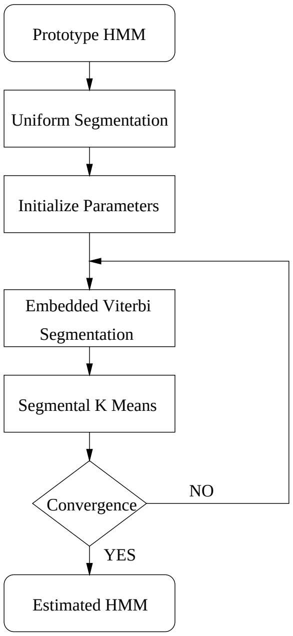

Figure 6: Training Scheme

1. According to the number of super states, the number of states in each super state, and the top-to-bottom and left-to-right structure of an embedded HMMprototype, the data is uniformly segmented to obtain initial estimates of the model parameters. First, the observations of the overall top-to bottom HMM are segmented in N_0 vertical super states, then, the data corresponding to each of this super states is uniformly segmented from left to right into $N_1^{(k)}$ states.

2. At the next iteration, the uniform segmentation is replaced by a doubly embedded Viterbi segmentation algorithm [10].

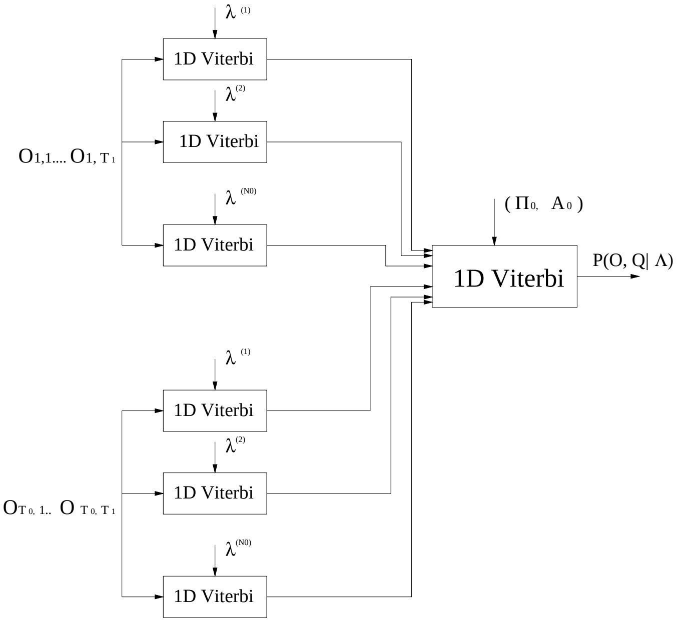

Figure 7: Doubly Embedded Viterbi Algorithm

The doubly embedded Viterbi segmentation algorithm, illustrated in Figure 7, consists of the following steps. First, the Viterbi segmentation is applied to each row of the image, and the probabilities

$$P(\mathbf{O}_{t_0,1} \ldots \mathbf{O}_{t_0,T_1}, q_{1,1}^{(t_0)} \ldots q_{1,T_1}^{(t_0)} | \lambda^{(k)})$$

$$1 \leq k \leq N_0$$

are calculated, where $q_{1,t_1}^{(t_0)}$, $1 \leq t_1 \leq T_1$ represent the state of a super state assigned to the observation $\mathbf{O}_{t_0,t_1}$. The probabilities of the states and observations in a row given the super state model, obtained from the Viterbi segmentation, represent the super state probabilities. The super state probabilities, together with the super state transition probabilities $\mathbf{A_0}$ and the initial super state probabilities $\mathbf{\Pi_0}$, are used to perform the Viterbi segmentation from the top to the bottom of the image and to determine:

$$P(\mathbf{O}_{1,1} \ldots \mathbf{O}_{1,T_1}, \ldots \mathbf{O}_{T_0,1} \ldots \mathbf{O}_{T_0,T_1}, q_{0,1} \ldots q_{0,T_0} | \lambda)$$

or using a shorthand notation $P(\mathbf{O}, \mathbf{Q}|\lambda)$. q_{0,t_0}, $1 \leq t_0 \leq T_0$ are the super states corresponding to row t_0.

3. The model parameters are estimated using an extension of the segmental k-means algorithm [12] to two dimensions. Therefore, the model parameters are obtained according to:

$$a_{1,ij}^{(k)} = \frac{number\ of\ transitions\ from\ S_{1,i}^{(k)}\ to\ S_{1,j}^{(k)}}{number\ of\ transitions\ from\ S_{1,i}^{(k)}}$$

$$\mu_i^{(k)} = \text{sample mean of vector } i \text{ in super state } k$$

$$\mathbf{U}_i^{(k)} = \text{sample covariance matrix of vectors in state } i \text{ of super state } k$$

$$a_{0,ij} = \frac{number\ of\ transitions\ from\ S_{0,i}\ to\ S_{0,j}}{number\ of\ transitions\ from\ S_{0,i}}$$

4. The iteration stop, and the HMM is initialized, when the Viterbi segmentation likelihood at consecutive iterations is smaller than a threshold.

5. FACE RECOGNITION

After extracting the observation vectors corresponding to the test face images, the probability of the observation sequence given an embedded HMM face model is computed via a doubly embedded Viterbi recognizer. The model with the highest likelihood is selected and this model reveals the identity of the unknown face.

The face recognition system has been tested on the Olivetti Research Ltd. database (400 images of 40 individuals, 10 images per individual at the resolution of 92 × 112 pixels). Half of the images were used in training, and the other half were used for testing. The database contains face images of people of different ages, both males and females, showing different facial expressions, hair styles, and eye wear (glasses/no glasses). On the same database, the recognition performance with a one-dimensional HMM [5], [8] was around 85%. The recognition rate of the "eigenfaces" method which depends on the number of eigenfaces used, varies from 73% with less than 5 eigenfaces to about 90% when 200 eigenfaces are used. The recognition performance for the pseudo 2-D HMM of Samaria [9] depends on the structure and the sampling that is used from 90% to 95%. However, due to the large dimension of the observation vectors, the system required about four minutes for a face to be recognized on Sparc 20 workstation.

The accuracy of the system presented in this paper is increased to 98% while the recognition time required for one face to be identified is significantly decreased compared to the structure presented in [9]. The efficiency of the system presented in this paper is due both to the choice of a more efficient observation vector and to the use of an efficient HMM structure. While it is obvious that a small size of the observation vector reduces the complexity of both the training and recognition the efficiency of the an embedded HMM-structure will be discussed in more detail. The number of additions required by the Viterbi decoder is N^2T, where N is the number of states and T is the number of observations. Let $NumAdds$ represent the number of additions required by the Viterbi decoder. Then, for the one-dimensional HMM

$$NumAdds = N^2T,$$

for Samaria's pseudo 2-D HMM [9]

$$NumAdds = (\sum_{k=1}^{N_0} N_1^{(k)})^2 T_0 T_1$$

and for an embedded HMM

$$NumAdds = (\sum_{k=1}^{N_0} (N_1^{(k)})^2 T_1) T_0 + {N_0}^2 T_0$$

Figure 8 presents some of the recognition results. The crossed images represent incorrect classifications, while the rest of images are examples of correct classification.

6. CONCLUSIONS

This paper describes an embedded HMM approach for face recognition that uses an efficient set of observation vectors based on the DCT coefficients. The use of an embedded HMM model for the human face is justified by the structure of the face, and is invariant for a large range of orientations, gestures, and face appearances. The use of an embedded HMM increases by over 10% the recognition rate of the one-dimensional HMM and the classical eigenfaces method. Furthermore, an embedded HMM can be used for the modeling of faces of different sizes without prescaling of the images (this is not possible for the template-based methods).

Future work will be directed towards building a face detection system that uses an embedded HMM face model. Further improvements of the an embedded HMMcan be obtained by using the state duration modeling and mixture density modeling of the states.

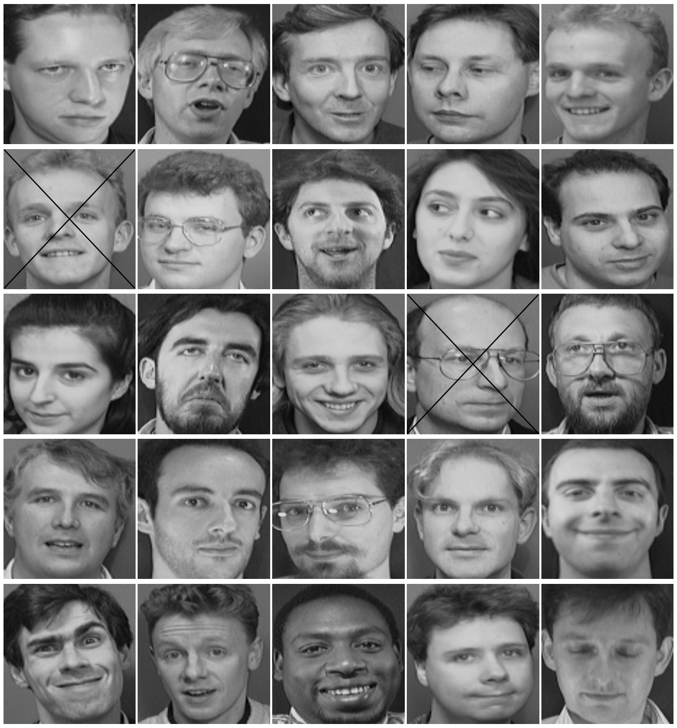

Figure 8: Face Recognition Results

7. REFERENCES

[1] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of IEEE*, vol. 83, May 1995.

[2] D. Beymer, "Face recognition under varying pose," in *Proceedings of 23rd Image Understanding Workshop*, vol. 2, pp. 837–842, 1994.

[3] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings of International Conference on Pattern Recognition*, pp. 586 – 591, 1991.

[4] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs Fisherfaces: Recognition using class specific linear projection," in *Proceedings of Fourth Europeean Conference on Computer Vision, ECCV'96*, pp. 45–56, April 1996.

[5] F. Samaria and S. Young, "HMM based arhitecture for face identification," *Image and Computer Vision*, vol. 12, pp. 537–543, October 1994.

[6] J. Ben-Arie and D. Nandy, "A volumetric/iconic frequency domain representation for objects with application for pose invariant face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 449–457, May 1998.

[7] A. V. Nefian and M. H. Hayes, "A Hidden Markov Model for face recognition," in *ICASSP 98*, vol. 5, pp. 2721–2724, 1998.

[8] A. V. Nefian and M. H. Hayes, "Face detection and recognition using Hidden Markov Models," in *International Conference on Image Processing*, 1998. to appear.

[9] F. Samaria, *Face Recognition Using Hidden Markov Models*. PhD thesis, University of Cambridge, 1994.

[10] S. Kuo and O. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-D Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 842–848, August 1994.

[11] L. Rabiner and B. Huang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of IEEE*, vol. 77, pp. 257–286, February 1989.