Graphical Models **62,** 323–342 (2000)
doi:10.1006/gmod.1999.0511, available online at http://www.idealibrary.com on IDEAL®

Locating Human Faces in a Cluttered Scene

A. N. Rajagopalan, K. Sunil Kumar, Jayashree Karlekar, R. Manivasakan, M. Milind Patil, U. B. Desai,[1] P. G. Poonacha, and S. Chaudhuri

Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Powai, Mumbai 400 076, India

Received March 30, 1998; revised August 11, 1999; accepted October 12, 1999;
published online July 1, 2000

In this paper, we present two new schemes for finding human faces in a photograph. The first scheme adopts a distribution-based model approach to face-finding. Distributions of the face and the face-like manifolds are approximated using higher order statistics (HOS) by deriving a series expansion of the density function in terms of the multivariate Gaussian and the Hermite polynomials in an attempt to get a better approximation to the unknown original density function. An HOS-based data clustering algorithm is then proposed to facilitate the decision process. The second scheme adopts a hidden Markov model (HMM) based approach to the face-finding problem. This is an unsupervised scheme in which face-to-nonface and nonface-to-face transitions are learned by using an HMM. The HMM learning algorithm estimates the HMM parameters corresponding to a given photograph and the faces are located by examining the optimal state sequence of the HMM. We present experimental results on the performance of both schemes. A training data base of face images was constructed in the laboratory. The performances of both the proposed schemes are found to be quite good when measured with respect to several standard test face images.

1. INTRODUCTION

The problem that we address in this paper is as follows: given a cluttered image, locate human faces in it. Although the task of recognizing a face has been well researched, that of locating a face in a complex or cluttered background is relatively unexplored. Face finding can serve as an important initial step toward building a fully automated face recognition system. It also finds applications in human–computer interfaces and surveillance/census systems. The task of face finding is difficult because there can be significant variations in the appearance of face patterns, which can be hard to parameterize analytically. The

[1] To whom correspondence should be addressed. E-mail: ubdesai@ee.iitb.ernet.in.

1524-0703/00 $35.00

problem is further compounded by unpredictable imaging conditions when the environment is unconstrained.

The first effort in face location was reported by Sakai *et al.* [1]. In their work, the model of a human face is defined in terms of several subtemplates that correspond to face-specific features. Fischler *et al.* [2] describe a framework of templates and springs for face location. The loose spatial relationships between different features of the face are modeled by the springs. In [3], Yuille *et al.* use deformable templates and an energy minimization approach for tackling the issue of scaling in face finding. Though template-based pattern matching techniques are simple, their performance has been found to be inadequate for satisfactory finding of faces.

In [4], Yang *et al.* present a scheme which is a three-level hierarchical knowledge-based pattern recognition system. At the top two levels, possible candidates are found according to certain face-finding rules while the third level performs edge finding to extract face characteristics. In [5], Sinha uses a small set of spatial image invariants between different parts of a human face to describe the space of face patterns. The scheme has been successfully demonstrated only on images with little background clutter. In [6, 7], Poggio and Sung describe an example-based learning approach for locating vertical frontal views of human faces in a cluttered image. In this scheme, a distribution-based model is developed for the frontal face views. A set of classification thresholds and parameters are learned for separating face and nonface patterns based on a set of distance measurements between the test pattern and the face model. The distance is measured using a two-value metric composed of the normalized Mahalanobis distance and the Euclidean distance. The scheme performs quite well even on cluttered images, but there is scope for improving the face-finding rate. In [8], Moghaddam and Pentland present a learning-based object-finding technique that is in some sense similar to [6]. The scheme models pattern classes by performing multimodal Gaussian density estimation using eigenvector decomposition. These probability densities are then used to formulate a maximum-likelihood estimation framework for finding faces in images. The approach has been demonstrated for images that contain only one face.

Early work on the use of neural networks for face location can be found in [9] and [10]. Though these methods are tolerant to scale and pose variances to some extent, their ability to handle multiple faces is not clear. Also, the schemes are error prone to high clutter and very close location of faces. Rowley *et al.* [11] report a highly constrained retinally connected multilayer perceptron net to classify image patterns. Their system arbitrates between the outputs of multiple networks for improved performance. Additional normalization procedures are introduced to generate face patterns in a principled fashion.

In a recent work, Govindaraju [12] presents a scheme for locating human faces in scenes under certain constraints. In the first stage, candidate regions are hypothesized by looking for gross features. These regions are next verified by searching for face-specific features. Edge contours are used as the basic features. The method is reported to be reliable with low false alarm rates. But there is room for refining the location of faces.

In this paper, we propose two new schemes for face finding. The first scheme is based on the higher order statistics (HOS) of the face and the face-like data samples while the second is an unsupervised scheme based on the hidden Markov model (HMM). Throughout this paper, the term “face-like” refers to nonface patterns as well as image patterns that look like faces but are actually not. A preliminary account of the proposed schemes can be found in [13].

In the HOS-based scheme, our approach is distribution-based as in [6], but there are important differences. We approximate the distribution of the face manifold using higher order statistics by deriving a series expansion of the density function in terms of the multivariate Gaussian function and the Hermite polynomials. The expansion uses higher order moments to get a better approximation to the original unknown distribution. The distribution of the face-like manifold is also modeled along similar lines so that the nonface regions near the face clusters can be carved out better. We believe that this refinement over a simple Gaussian fit can lead to an improved performance. We also propose a clustering algorithm that makes use of an HOS-based decision measure for better discriminating capability. We would like to emphasize that we do not perform any preprocessing operations on the face and the face-like patterns. The size of our face window pattern is relatively smaller than those used in [6] and [11], and our training data set is also quite small. Yet, the performance of our HOS-based scheme is found to be quite good even on images with multiple faces in a complex background.

The second approach that we propose here is based on the concept of *learning* the face-to-nonface and nonface-to-face transitions in a photograph. The hidden Markov Model is well tailored for this problem as it can learn state transitions given an observation sequence. The approach is based on generating an observation sequence from the photograph and learning the HMM parameters corresponding to this sequence. The postprocessed optimal state sequence decides the location of faces in the photograph. Our HMM-based approach can be termed "unsupervised" in the sense that we have no *a priori* knowledge as to which subimages in the photograph correspond to faces since the given photograph is our training data as well as test data. The HMM-based scheme also locates faces successfully even in a cluttered image.

The organization of the paper is as follows. In Section 2, we derive the multivariate series expansion of a probability density function in terms of the multivariate Gaussian and the Hermite polynomials. The HOS-based face finding scheme is described in Section 3. In Section 4, we discuss the HMM-based scheme for face finding. Experimental results are presented in Section 5 while Section 6 concludes the paper.

2. MULTIVARIATE SERIES EXPANSION

In this section, we derive a series expansion for a multivariate density function in terms of the Gaussian function and the Hermite polynomials. The corresponding expansion for the univariate case is quite well known [14, 15]. In Section 2.1, we derive some important orthogonality relations. These relations are then used in Section 2.2 to derive the coefficients in the series expansion of the multivariate probability density function (p.d.f). It may be noted that one can derive an HOS-based decision measure using this expansion. This measure can then be used to replace the Euclidean or the Mahalanobis distance for better discriminating capability in data clustering.

2.1. *Orthogonality Relations*

Let the random vector $X = [X_1 \;\; X_2 \;\; \cdots \;\; X_N]^T$ and $X \sim N(0, I)$. If $t = [t_1 \;\; t_2 \;\; \cdots \;\; t_N]^T$, then the moment generating function of X is given by

$$\Phi(t) = E[\exp(t^T X)].$$

Since these random variables are statistically independent, $\Phi(\underline{t}) = \exp(\frac{1}{2}\underline{t}^T\underline{t})$. Therefore, $E[\exp(\underline{t}^T X - \frac{1}{2}\underline{t}^T\underline{t})] = 1$. Replacing $\underline{t}$ by $\underline{t} + \underline{s}$ in this equation, we get

$$E\left[\exp\left(\underline{t}^T X - \frac{1}{2}\underline{t}^T\underline{t}\right)\exp\left(\underline{s}^T X - \frac{1}{2}\underline{s}^T\underline{s}\right)\right] = \exp(\underline{t}^T\underline{s}). \tag{1}$$

We know that a scalar valued multidimensional function $g(x)$ can be expanded using the Taylor series as

$$g(x) = \sum_{n=0}^{\infty} \frac{1}{n!}(x^{\otimes n})^T\left[\left(\underline{D}_x^{\otimes n}\right)g(x)\right]_{x=0},$$

where

$$x^{\otimes n} = \underbrace{x \otimes x \otimes \cdots \otimes x}_{n \text{ times}},$$

and $\otimes$ represents a tensor product and

$$\underline{D}_x = \left[\frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \cdots \quad \frac{\partial}{\partial x_N}\right]^T.$$

Therefore, we can write

$$\exp\left(\underline{t}^T x - \frac{1}{2}\underline{t}^T\underline{t}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\underline{t}^{\otimes n})^T \bar{H}_n(x), \tag{2}$$

where the Taylor series expansion is on the variable t. In (2), the vector $\bar{H}_n(x)$ is given by $\bar{H}_n(x) = [(\underline{D}_t^{\otimes n})\exp(\underline{t}^T x - \frac{1}{2}\underline{t}^T\underline{t})]_{\underline{t}=0}$ and $\underline{D}_t = [\frac{\partial}{\partial t_1} \ \frac{\partial}{\partial t_2} \ \cdots \ \frac{\partial}{\partial t_N}]^T$. Using (2) in (1) and expanding $\exp(\underline{t}^T\underline{s})$ also in Taylor series, we obtain

$$E\left[\sum_{m,n=0}^{\infty}(\underline{t}^{\otimes n})^T \frac{\bar{H}_n(X)}{n!}\frac{\bar{H}_m^T(X)}{m!}(\underline{s}^{\otimes m})\right] = \sum_{n=0}^{\infty}\frac{1}{n!}(\underline{t}^{\otimes n})^T(\underline{s}^{\otimes n}) = \sum_{m,n=0}^{\infty}\frac{1}{n!}(\underline{t}^{\otimes n})^T I_{q_n q_m}(\underline{s}^{\otimes m}), \tag{3}$$

where

$$\begin{aligned} I_{q_n q_m} &= O_{q_n q_m} \qquad \text{for } n \neq m \\ &= I_{q_n} \qquad \text{for } n = m. \end{aligned}$$

Here $O_{q_n q_m}$ is the zero matrix with q_n rows and q_m columns while I_{q_n} is the identity matrix of dimension q_n. The dimensions of the vectors $\overline{H_n}(x)$ and $\overline{H_m}(x)$ are given by q_n and q_m, respectively.

By equating the coefficients of $\underline{t}$ and $\underline{s}$ on both sides of (3), we obtain the important orthogonality relation

$$E\left[H_n(X)H_m^T(X)\right] = I_{p_n p_m}. \tag{4}$$

Note that the expectation is w.r.t $N(\underline{0}, I)$. In (4), $H_n(\underline{x})$ is a vector whose elements are given by the product

$$\left(\prod_{i=1}^{N} \frac{H_{k_i}(x_i)}{\sqrt{k_i!}}\right)$$

for all permutations of $k_i, i = 1, \ldots N$, such that $\sum_{i=1}^{N} k_i = n$. The dimensions of the vectors $H_n(\underline{x})$ and $H_m(\underline{x})$ are given by p_n and p_m, respectively. The term $H_{k_i}(x_i)$ is the Hermite polynomial of order k_i and is defined as

$$\left[\frac{\partial^{k_i}}{\partial t_i^{k_i}} \exp\left(t_i x_i - \frac{1}{2} t_i^2\right)\right]_{t_i=0}.$$

For example, if $N = 2$, $n = 3$ and $m = 2$, then

$$H_3(x_1, x_2) = \left[\frac{H_3(x_1)}{\sqrt{3!}} \;\vdots\; \frac{H_2(x_1)}{\sqrt{2!}} \frac{H_1(x_2)}{\sqrt{1!}} \;\vdots\; \frac{H_1(x_1)}{\sqrt{1!}} \frac{H_2(x_2)}{\sqrt{2!}} \;\vdots\; \frac{H_3(x_2)}{\sqrt{3!}}\right]^T,$$

$$H_2(x_1, x_2) = \left[\frac{H_2(x_1)}{\sqrt{2!}} \;\vdots\; \frac{H_1(x_1)}{\sqrt{1!}} \frac{H_1(x_2)}{\sqrt{1!}} \;\vdots\; \frac{H_2(x_2)}{\sqrt{2!}}\right]^T, \qquad p_3 = 4 \quad \text{and} \quad q_2 = 3.$$

Further, $H_0(x) = 1$, $H_1(x) = x$, $H_2(x) = x^2 - 1$ and $H_3(x) = x^3 - 3x$.

It is also possible to derive an orthogonality relation in terms of $N(\underline{\mu}, R)$. It will be shown later that this relation has advantages when the mean and the covariance of $\underline{X}$ are $\underline{\mu}$ and R, respectively. From (4), we have

$$\int \frac{1}{(2\pi)^{\frac{N}{2}}} \exp\left(-\frac{1}{2} \underline{x}^T \underline{x}\right) H_n(\underline{x}) H_m^T(\underline{x})\, d\underline{x} = I_{p_n p_m}.$$

The above integral is an N-fold integral whose limits are $-\infty$ and ∞. Let $\underline{x} = R^{-\frac{1}{2}}(\underline{y} - \underline{\mu})$, where $R^{\frac{1}{2}} = UD^{\frac{1}{2}}U^T$. Matrices U and D can be obtained through an eigenvalue-eigenvector decomposition of the covariance matrix R. Since, $d\underline{x} = (\det R^{-\frac{1}{2}})\, d\underline{y}$, we get

$$\int \frac{1}{(2\pi)^{\frac{N}{2}} (\det R)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\underline{y} - \underline{\mu})^T R^{-1}(\underline{y} - \underline{\mu})\right) H_n\left(R^{-\frac{1}{2}}(\underline{y} - \underline{\mu})\right)$$
$$\times H_m^T\left(R^{-\frac{1}{2}}(\underline{y} - \underline{\mu})\right) d\underline{y} = I_{p_n p_m}.$$

Thus, we obtain the orthogonality relation

$$E\left[H_n\left(R^{-\frac{1}{2}}(\underline{X} - \underline{\mu})\right) H_m^T\left(R^{-\frac{1}{2}}(\underline{X} - \underline{\mu})\right)\right] = I_{p_n p_m}, \tag{5}$$

where the expectation is now w.r.t $N(\underline{\mu}, R)$.

2.2. *Expansion of a Multivariate p.d.f*

Let $f(\underline{x})$ be a multivariate probability density function. We wish to expand $f(\underline{x})$ as

$$f(\underline{x}) = N(\underline{0}, I) \left(\sum_{m=0}^{\infty} C_m H_m(\underline{x})\right). \tag{6}$$

Multiplying both sides of (6) by $H_n^T(x)$ and integrating, we get

$$\int f(x)\, H_n^T(x)\, dx = \sum_{m=0}^{\infty} C_m \int N(0, I) H_m(x) H_n^T(x)\, dx = C_n I_n. \quad \text{(from (4))}$$

Therefore,

$$C_n = \int f(x) H_n^T(x)\, dx = E\left[H_n^T(X)\right].$$

Note that the expectation is w.r.t $f(x)$. Thus, $f(x)$ can be written as

$$f(x) = N(0, I)\left(1 + \sum_{n=1}^{\infty} E\left[H_n^T(X)\right] H_n(x)\right).$$

Similarly, $f(x)$ can also be expanded in terms of $N(\underline{\mu}, R)$ using the relation derived in (5) as

$$f(x) = N(\underline{\mu}, R)\left(1 + \sum_{n=1}^{\infty} E\left[H_n^T\left(R^{-\frac{1}{2}}(X - \underline{\mu})\right)\right] H_n\left(R^{-\frac{1}{2}}(x - \underline{\mu})\right)\right).$$

The advantage of this expansion is that when the mean and the covariance of X are $\underline{\mu}$ and R, respectively, then the above equation simplifies to

$$f(x) = N(\underline{\mu}, R)\left(1 + \sum_{n=3}^{\infty} E\left[H_n^T\left(R^{-\frac{1}{2}}(X - \underline{\mu})\right)\right] H_n\left(R^{-\frac{1}{2}}(x - \underline{\mu})\right)\right). \tag{7}$$

Note that the summation goes from 3 and onward. In the context of face detection, the random vector X represents the class of face/nonface images while x is a lexicographically ordered sample image pattern from the class.

3. HOS-BASED FACE FINDING

In this section, we propose an HOS-based face finding scheme that finds faces by searching an image for square patches of frontal views of the human face. The search for these face window patterns is carried out at all points of the image and across different scales. At each location and scale, the system classifies a window pattern as being either a face or a nonface based on a set of difference measurements between the pattern and the face model. The difference is computed using an HOS-based measure. We now discuss face modeling, data clustering, and face classification in the HOS-based face finding scheme.

3.1. *Face Modeling*

Analogous to [6], we use a distribution-based model for face finding. In this scheme, canonical faces are represented as the set of all masked 13×13 patterns that are canonical face views (See Appendix A for details on masking). The size of the window has been chosen to be 13×13 pixels as a compromise between computational complexity and preservation of structural details of face patterns. The upper boundary lies just above the eyes while the lower edge is just below the mouth. The class of all masked 13×13 images forms a vector space. The dimension of this space equals 131 in our scheme. This is the number of image pixels that are retained after masking. The set of all 13×13 pixel canonical face

patterns maps to a manifold in this multidimensional vector space. If this manifold can be exactly recovered, then one can model the class of all canonical face views. But the global shape of this manifold may be quite complex and analytically intractable. Hence, one must approximate the underlying distribution. A single multidimensional cluster may be a poor description of the space of canonical face views. This is because a single cluster can result in a significant amount of overlap between the face and the face-like pattern distributions and may not separate the two pattern classes adequately enough. Hence, one needs to approximate the face pattern distribution in a piecewise-smooth fashion. This would be a reliable representation as long as the actual distribution is locally linear [6].

The actual density $f(\underline{x})$ of the face patterns is approximated up to its mth order joint moment by the function $\hat{f}(\underline{x})$ which is given by

$$\hat{f}(\underline{x}) = N(\underline{\mu}, R)\left(1 + \sum_{n=3}^{m} E\left[H_n^T\left(R^{-\frac{1}{2}}(X - \underline{\mu})\right)\right] H_n\left(R^{-\frac{1}{2}}(\underline{x} - \underline{\mu})\right)\right). \tag{8}$$

It may be mentioned here that in [6] the distribution of the face manifold was modeled by fitting the face data sample with multidimensional Gaussian clusters. The clusters were obtained by fitting mean vectors and covariance matrices to the face sample distribution. It is our conjecture that the face pattern distribution is unlikely to be governed by a simple multidimensional Gaussian function. Hence, this is the motivation to approximate $f(\underline{x})$ by (8) which incorporates higher order moments to get a better approximation than a simple Gaussian fit. Under the piecewise-smoothness assumption of the actual manifold, we too use (as in [6]) six clusters to describe the face distribution. The distribution of the face manifold is modeled by fitting the face data samples with six multidimensional clusters, each consisting of a centroid, a covariance matrix, and joint higher order moments up to order m. It must be noted here that all the moments in (8) must be *estimated* from the face data samples. The computational complexity depends on the value of m, the order of the Hermite polynomial. In our experiments, the value of m was chosen to be 3 as a compromise between accuracy of representation of $f(\underline{x})$ and computational complexity.

In the real world, many naturally occurring nonface patterns look like faces when viewed in isolation. It is possible that some of these face-like patterns may be located quite close to the centroids of the face clusters. This can result in misclassification because the face patterns are expected to lie nearer to the face cluster centroids than nonface patterns. Hence, it is desirable to refine the manifold boundary by explicitly carving out regions around the face sample distribution that do not correspond to canonical face views. Toward this end, we model the distribution of the face-like manifold also by (8), the basic argument again being the same as that advocated earlier for the face patterns. This is unlike in [6], where the distribution of the face-like manifold is modeled by a Gaussian function. The exercise that was undertaken in obtaining an approximation to the face pattern distribution is now repeated for the face-like patterns. We believe that by thus modeling the distributions of the face and the face-like manifolds, we should be able to better carve out the nonface patterns near the face clusters.

3.2. *Data Clustering*

In the traditional k-means algorithm, the Euclidean and the Mahalanobis distances are used as similarity measures for clustering purpose [16, 17]. Since the clustering is based only on the mean and the second order statistics, the underlying assumption is that the distribution of the data is Gaussian. If the actual distribution of the data is non-Gaussian,

then traditional k-means algorithms may fail to yield satisfactory results. This is therefore the motivation to develop a clustering algorithm that would use a decision measure that incorporates higher order (>2) statistics for improved clustering. Also, it would be nice if the Mahalanobis metric turns out to be a special case of such a decision measure when the underlying distribution is Gaussian.

We define the HOS-based decision measure as $(-\log f(\underline{x}))$. If the mean and the covariance of $\underline{X}$ are $\underline{\mu}$ and R, respectively, then from (8) the *HOS-based finite order decision measure* is given by

$$-\log\left(N(\underline{\mu}, R)\left(1+\sum_{n=3}^{m} E\left[H_n^T\left(R^{-\frac{1}{2}}(\underline{X}-\underline{\mu})\right)\right] H_n\left(R^{-\frac{1}{2}}(\underline{x}-\underline{\mu})\right)\right)\right). \tag{9}$$

We observe that when the density function is non-Gaussian, the above measure takes the higher order moments (>2) also into consideration and this is what lends better discriminating power to it. Inclusion of the higher order moments does not increase the computational complexity prohibitively because, as can be noted, there is no inversion operation associated with the higher order moments. Interestingly, when $f(\underline{x})$ is Gaussian, the HOS-based decision measure neatly reduces to the normalized Mahalanobis distance because all the terms in the summation go to zero. We now proceed to describe the HOS-based clustering algorithm that is used for performing data clustering in the proposed face finding scheme.

3.2.1. HOS-based clustering algorithm.

1. Obtain k initial pattern centers with Euclidean distances on the image database (either face or nonface). Divide the data set into k clusters by assigning each data sample to the nearest pattern center in Euclidean space.
2. Initialize the joint moments (second and onwards up to order m) of all k clusters.
3. Recompute pattern centers to be the centroids of the current data partitions.
4. Using the current set of k pattern centers and their higher order moments, recompute data partitions by reassigning each data sample to the nearest pattern center using the HOS-based decision measure defined in (9). If the data partitions remain unchanged or if the maximum number of inner-loop (i.e., steps 3 and 4) iterations have been exceeded, proceed to step 5. Otherwise, return to step 3.
5. Recompute the moments (second and onward up to order m) of all k clusters from their respective data partitions.
6. Using the current set of k pattern centers and their cluster moments, recompute data partitions by reassigning each data sample to the nearest pattern center using the HOS-based measure. If the data partitions remain unchanged or if the maximum number of outer loop (i.e., steps 3 to 6) iterations have been exceeded, proceed to step 7. Otherwise, return to step 3.
7. Return the current set of k pattern centers and their joint moments (up to order m), for each cluster.

It may be noted that in our experiments, the value of k was 6 and that of m was 3.

3.3. *Face Classification*

Though it may seem that detection could be simply done by checking the p.d.f.s for the face and nonface classes, it is important to note that the boundaries of the clusters

are not sharp enough [6]; there are "fuzzy" regions where one could not categorically say whether the given image is a face or a nonface. A multilayer perceptron (MLP) net is used to demarcate the boundaries better by learning a set of classification thresholds for locating face patterns. The neural net-based classifier learns the functional mapping from input feature distance vectors to output classes using a representative set of training examples.

The network has 12 input nodes, 6 hidden nodes, and 1 output node. The parameters of the network are learnt for classifying face and nonface patterns. The input vector to the network consists of 12 elements that are the HOS-based finite order decision measure of the image pattern with respect to the 12 (6 face and 6 face-like) clusters of the model. These are computed using the HOS-based measure given in (9). The net is trained until the output error stabilizes to a small value. The output node returns a 1 for face patterns and a 0 otherwise.

Once training is complete, to find whether a given image pattern is a face or not, each candidate pattern is matched against the distribution-based face and face-like models; i.e., a 12-dimensional vector of the HOS-based decision measure corresponding to the test pattern is computed using (9). This vector is then fed to the neural net-based classifier. The trained classifier determines whether or not the test pattern is a face. It may be noted that this is unlike in [6], where a 24-element vector of difference measurements is used for training and finding. In [6], the first distance component measures the partial distance between the test pattern and the cluster's 75 most significant eigenvectors, while the second distance component is the Euclidean distance between the test pattern and its projection in the 75-dimensional subspace.

We mention here that when the underlying distributions of the face and the face-like manifolds are assumed to be Gaussian and the HOS-based decision measure is replaced by the Euclidean/Mahalanobis distance measure for data clustering, then one obtains the Euclidean/Mahalanobis-based face finder.

4. HMM-BASED FACE FINDING

The idea of using a hidden Markov model (HMM) for locating faces is essentially based on being able to fit the face finding problem into the HMM framework. In this approach, the underlying concept is to capture the face-to-nonface, and nonface-to-face transitions in a given image using an HMM. A given photograph is raster scanned to generate an observation sequence. Having determined the number of states of the HMM *a priori* and ensured that at least one state corresponds to face, the observation sequence is used to find the HMM parameters. The optimal state sequence of the HMM then decides the location of faces in the image. It may be mentioned that HMM has been previously used, among others, by Samaria and Young for face *recognition* [18].

In this section, we first give a brief introduction to HMM. A detailed discussion can be found in [19, 20]. The HMM-based scheme for face finding is discussed next.

4.1. *Hidden Markov Models*

A hidden Markov model is a doubly stochastic process manifesting through an observed sequence of feature vectors. The HMM consists of a finite number of states whose transitions are based upon a transition probability distribution which is Markovian. In a particular state, an observation is produced according to a probability distribution which depends on the current state. It is only the observations that are visible; the states are "hidden."

An HMM is characterized by the following parameters.

- The number of states $\mathcal{N}$.
- An observation sequence consisting of symbols.
- The state transition probability matrix A whose ijth element represents the probability of transiting to state j from state i.
- The probability matrix B whose ikth element represents the probability of selecting symbol k from state i.
- The vector π whose ith element gives the probability of starting in state i.

For compact notation, the HMM parameters are denoted by $\lambda = (A, B, \pi)$. The HMM parameters must be learned from the given observations.

HMM learning algorithm. 1. Generate $\mathcal{N}$ clusters from the observation sequence. This also gives the initial state sequence.

2. Compute the probabilities A, B, and π using the empirical definition of probability.

3. Determine the optimal sequence using Viterbi algorithm [21] and HMM parameters.

4. If the optimal state sequence matches with the current state sequence, then stop (HMM learning is complete). Else re-assign symbols and go to step 2.

4.2. *Locating Faces Using HMM*

The motivation for using HMM for face finding comes naturally once it is recognized that the raster scanned subsets of the photograph (see Fig. 1) form the observation sequence of the HMM and that the face finding problem would then effectively boil down to identifying a subset of the photograph as a face or nonface in the optimal state sequence. The vast success that HMM has enjoyed in the area of speech recognition has been another motivating factor for using HMM.

The face finding scheme using HMM can be broadly divided into three sections, viz. a preprocessing section, a processing section, and a postprocessing section. In the preprocessing section, given an image, the observation sequence that the HMM must learn is first generated. We generate this sequence in the transform domain by comparing each masked subimage with a knowledge-base consisting of 6 face and 6 face-like centroids. In the processing section, the number of states to be used for the HMM is first decided. Since the nonfaces in a photograph typically outnumber the faces, it must be ensured that at least one state of the HMM represents the face. The k-means algorithm is then used to cluster the observation sequence into $\mathcal{N}$ states. The HMM is then trained on this sequence. After learning is complete, the postprocessor first binarizes the optimal state sequence. This is

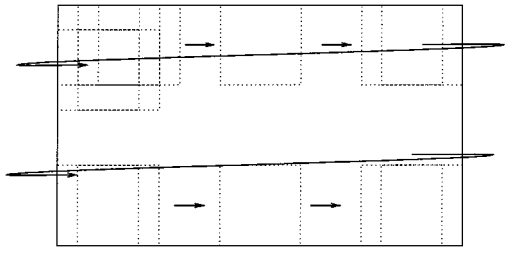

FIG. 1. A raster scanned photograph.

followed by a "cleaning" operation to remove the background clutter. Finally, the postprocessed optimal state sequence gives the location of faces in the image. We now describe each section of the scheme in detail.

4.2.1. Preprocessing. 1. Construct $m \times n$ subimages, $\{A_{i,j}\}_{i,j=m/2,n/2}^{M-m/2,N-n/2}$, by traversing the photograph of size $M \times N$ in a raster-scan fashion as shown in Fig. 1. Crop each subimage with a mask as discussed in Appendix A. The cropped subimages are then lexicographically ordered as column vectors $\{A'_{i,j}\}_{i,j=m/2,n/2}^{M-m/2,N-n/2}$.

2. Construct a 12×1 vector

$$\Delta_{i,j} = \left[\delta_{i,j}^1, \delta_{i,j}^2, \ldots, \delta_{i,j}^{12}\right]^T = [\|A'_{i,j} - K_1\|^2, \|A'_{i,j} - K_2\|^2, \ldots, \|A'_{i,j} - K_{12}\|^2]^T$$

using the face centroids $K_1, K_2, \ldots, K_6$ and the face-like centroids $K_7, K_8, \ldots, K_{12}$ from the knowledge base. Though we use the Euclidean norm for its simplicity, one could also use the Mahalanobis or the HOS-based measure to generate the observation sequence in the transform domain. The knowledge-base should then contain the appropriate second and higher order statistics.

3. $\{\Delta_{i,j}\}_{i,j=m/2,n/2}^{M-m/2,N-n/2}$ is now the observation sequence that the HMM must learn.

4.2.2. Processing. 1. Determine the number of states $\mathcal{N}$ to be used for the HMM as follows:

(a) Initialize $\eta = 2$ and select $\eta_{\max}$.

(b) Cluster $\{\Delta_{i,j}\}_{i,j=m/2,n/2}^{M-m/2,N-n/2}$ into η bins using k-means clustering algorithm.

(c) Determine the centroids of each of the η bins (say$\{{}^k\Delta\}_{k=1}^{\eta}$, where ${}^k\Delta$ is a vector of length 12, namely, $[{}^k\delta^1, {}^k\delta^2, \ldots, {}^k\delta^{12}]$).

(d) for $k = 1$ to η, compute

$$\alpha_* = \arg\min_{\alpha} {}^k\delta^{\alpha} \tag{10}$$

If for some $k = k^*$, (10) returns an α_* such that $1 \le \alpha_* \le 6$, then the bin corresponding to k^* represents a *face*. Note that δ's represent the distances from the knowledge base (face$\{K_1, \ldots, K_6\}$, nonface $\{K_7, \ldots, K_{12}\}$). Output $\mathcal{N} = \eta$ as the number of states to be used by HMM, else $\eta = \eta + 1$, until $\eta_{\max}$. If $\eta_{\max}$ is reached without (10) being satisfied then there is no cluster corresponding to a *face* in the photograph—`STOP` the algorithm and decide that there are no faces in the photograph.

2. Use the k-means clustering algorithm to cluster $\{\Delta_{i,j}\}$ into $\mathcal{N}$ states and to determine the initial state sequence $\{\mathcal{S}_{i,j}\}$, $\mathcal{S}_{i,j} \in (1, \mathcal{N})$, corresponding to each observation $\Delta_{i,j}$.

3. HMM training:

(a) Using the observation sequence $\{\Delta_{i,j}\}$ and the initial state sequence $\{\mathcal{S}_{i,j}\}$ determine the HMM parameters (λ) as

$$\max_{\lambda} \mathcal{P}[\{\Delta_{i,j}\}, \{\mathcal{S}_{i,j}\} \mid \lambda]$$

(b) the optimal state sequence $\{\mathcal{S}_{i,j}^0\}$ is next obtained using Viterbi algorithm [21] for iter $= 1$ to `ITER` (`ITER` predefined)

(i) $\max_{\lambda} \mathcal{P}[\{\Delta_{i,j}\}, \{\mathcal{S}_{i,j}^{\text{iter}}\} \mid \lambda]$ to determine λ.

(ii) using λ, determine the optimal state sequence $\{\mathcal{S}^{\text{iter}+1}\}$

(iii) if $\{\mathcal{S}_{i,j}^{\text{iter}}\}$ and $\{\mathcal{S}_{i,j}^{\text{iter}+1}\}$ are identical, namely the state sequence at two consecutive iterations do not change—STOP and output $\{\mathcal{S}_{i,j}^{*}\}$ as the optimal state sequence, else iter = iter + 1.

It may be mentioned here that the segmental k-means algorithm described above yields HMM parameters λ^{k+1} from λ^{k} such that $P(\{\Delta_{i,j}\}, \{S_{i,j}^{k+1}\}^* \mid \lambda^{k+1}) \geq P(\{\Delta_{i,j}\}, \{S_{i,j}^{k}\}^* \mid \lambda^{k})$. The optimal state sequence thus obtained is a refinement over the initial state sequence.

4.2.3. Postprocessing.

1. This consists of binarizing the optimal state sequence $\{\mathcal{S}_{i,j}^{*}\}$ as follows:
   (a) determine the state corresponding to face.
   (b) set the state corresponding to face to 1 (say) and all the other states to 0 (say).
   (c) the state corresponding to face is to be processed and the rest are assumed to correspond to background.
2. Apply morphological operations *erosion* and *dilation* on the binarized $\{\mathcal{S}_{i,j}^{*}\}$ as explained in Appendix B.

Use the processed final state sequence to mark *faces* in the original photograph.

5. EXPERIMENTAL RESULTS

In this section, we give a comparison of the performance of the proposed HOS-based and HMM-based face finding systems versus the Euclidean-based face finder and the systems proposed in [6] and [11]. The training set consisted of 2004 "face" patterns, 4065 "face-like" patterns, and 6364 additional "nonface" patterns. The same training set was used for the Euclidean-based system. Details on data acquisition are given in Appendix A. It is important to note here that the size of our training set is much smaller than the one in [6], where 4150 face patterns, 6189 face-like patterns, and about 43,000 nonface patterns are used to train the neural net.

The test data base reported in [11] was used for comparison of performance. This set is completely distinct from our training set. The test data base consists of several face views (mostly frontal) of different people. The images consist of both single and multiple faces and include hand-drawn faces also. Because the images in this data base vary considerably in quality, the data base serves as a good measure to evaluate the sensitivity of performance of the face-finding system to the quality of images. Moreover, as the data base also includes images with a complex background, it serves as a reliable test for how well the system rejects nonface patterns. Each image in the test data base is scanned for square patches of frontal face views of the human face. The search for a face is carried out at all points in the image and across different scales. If a face window pattern is found by the system, an appropriate sized box is drawn at the corresponding window location in the output image. Output results are given for both the HOS-based and the HMM-based face finding schemes.[2] For the purpose of comparison, a quantitative breakdown of the performance of each of the above schemes is tabulated over the same test data base.

Results of HOS-based scheme. The size of the canonical face pattern used in our scheme is only 13×13 pixels which is quite small as compared to the window size of 19×19 pixels

[2] Detection results corresponding to some of the test images in [11, 13] have not been shown here but can be viewed at http://www.ee.iitb.ernet.in/~ ubdesai/GMIP_00/Results/face_detection.html.

used in [6] and 20×20 pixels in [11]. We do not carry out any preprocessing operations on the face data set unlike in [11], where each original face pattern is normalized so that both the eyes appear on a horizontal line and the image is scaled so that the distance from a point between the eyes to the upper lip is fixed for all images.

Figure 2 shows the output results of the HOS-based face finding system on the test data base. Figures 2A–2C consist of a single face against a simple background. The system finds the faces without any false matches. Figures 2D–2H contain a single face image in a

FIG. 2. Output results of the HOS-based face finding scheme. Below every image, three values are given: These correspond to the number of faces in the image, the number of faces correctly found, and the number of false alarms, respectively.

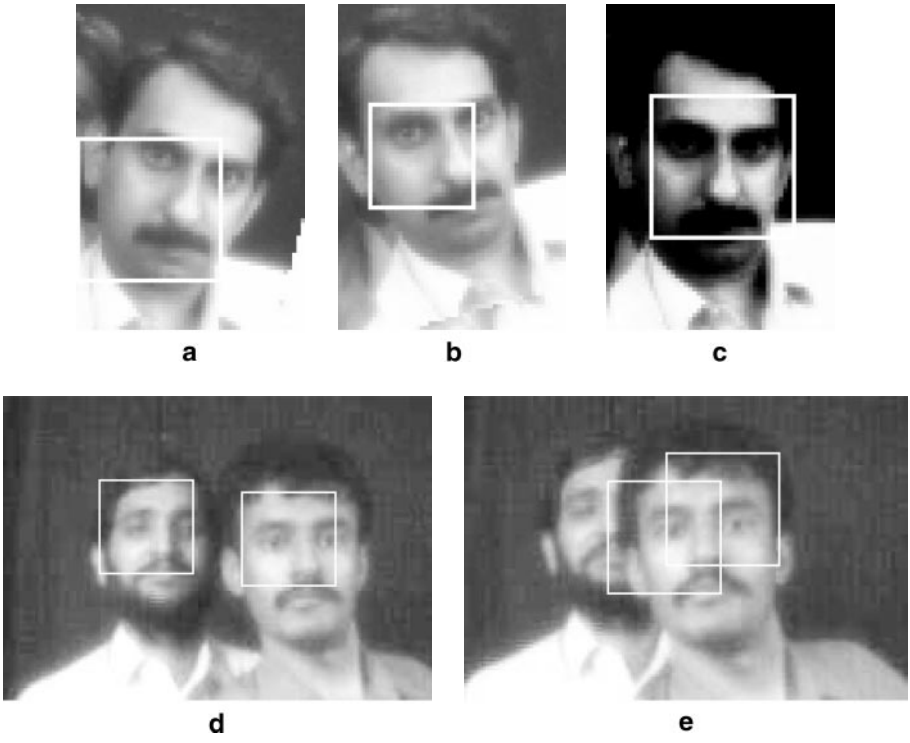

FIG. 3. Detection results for the HOS-based method under (a) and (b) image rotation, (c) shadowing, and (d) and (e) occlusion.

complex background. Although, all the faces are correctly found, there is one false alarm in Fig. 2F. Interestingly, the system successfully finds the face in the noisy Fig. 2I too. It may be noted that the system was not trained for hand-drawn or noisy faces at all. Figures 2J–2K contain multiple face images against a complex background while Fig. 2L is an image of playing cards. Again, we observe that most of the faces are found (only one face is missed in Fig. 2L). However, Fig. 2L contains a few false alarms. In comparison, we mention here that in [11] one face is missed in Fig. 2K while as many as three faces are missed in Fig. 2L.

Experiments were also conducted to test the sensitivity of the HOS-based method to different degrees of rotation, shadowing, and occlusion. Some of the detection results are shown in Fig. 3.

From the figure, it is clear that the method is reasonably robust to rotation. Our experiments reveal that the method can tolerate about 10 to 15° of rotation. Also, the method is reasonably robust to shadowing effects. As mentioned earlier, the method is trained on that region of the face whose upper boundary lies just above the eyes and whose lower edge falls just below the mouth. If there is occlusion in this area, the method may fail depending upon the severity of the problem. As shown in Fig. 3, the method fails when one of the faces is severely occluded (such as by another face in the front). Occlusion/shadowing in other areas of the face do not pose any problems.

Thus, we observe that the HOS-based scheme is able to locate faces quite well in all the images in the test data base. The system works reliably, even for fairly complicated scenes. However, our false alarms tend to be marginally more.

Results of HMM-based scheme. Experiments were next carried out to validate the HMM-based face finding scheme. The scheme was tested on the same test data base that was used for the HOS-based scheme. The knowledge-base for the HMM scheme was obtained by clustering the face and the face-like data samples in our training set using the k-means algorithm. The centroids of these clusters were then used to calculate the Δ_{ij}s as described in Section 4.2.1 using the Euclidean norm.

Figure 4 shows the output results of the HMM-based face finding scheme on the test data base. In most of the photographs with a single face, the scheme located faces correctly

FIG. 4. Results of the HMM-based face finding scheme for the images in the test data base. The values inside each box represent the same quantities as in Fig. 2.

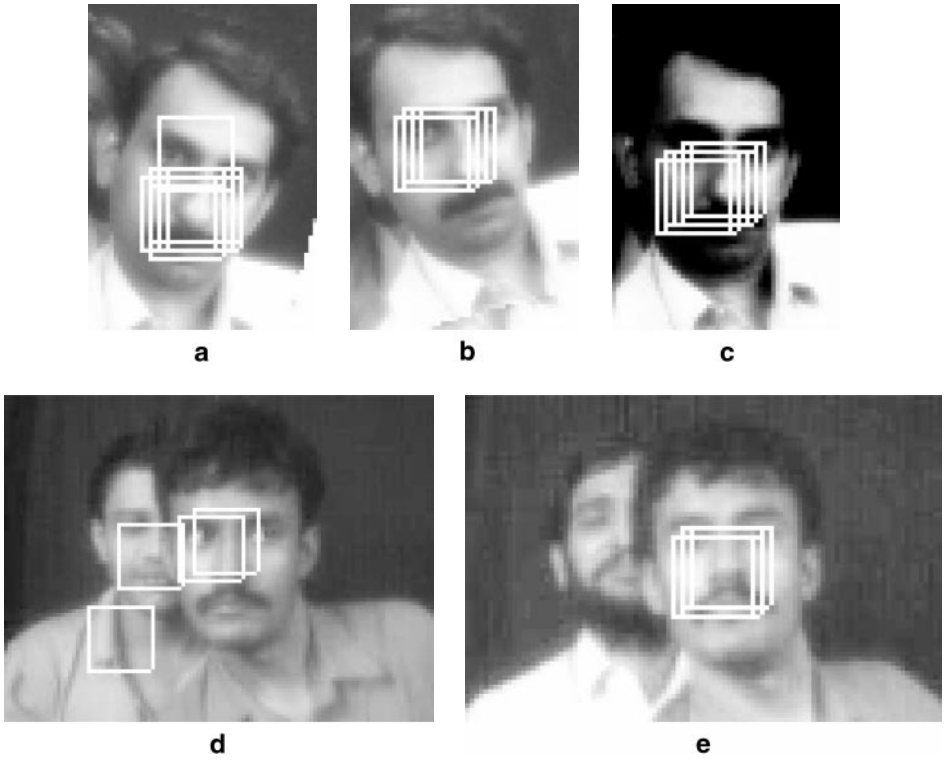

FIG. 5. Detection results for the HMM-based method under (a) and (b) image rotation, (c) shadowing, and (d) and (e) occlusion effects.

without any false alarms. In Figs. 4F and 4H, though the faces are located, there are a few false matches due to background clutter. The system could correctly locate the face in the noisy Fig. 4I. The scheme performs quite well on photographs with multiple faces too. In Fig. 4J we are able to detect all the faces. In Fig. 4K, we are able to locate all the faces as in the HOS-based scheme. In comparison, it may be noted that the scheme proposed in [11] could capture only three faces in Fig. 4K. In Fig. 4L, we detect six faces but there are a few false alarms.

Finally, the HMM-based method was also tested against different degrees of rotation, occlusion, and shadowing in the images. The detection results are shown in Fig. 5. Clearly, the method is reasonably robust to the above effects. The performance of the HMM-based method was again found to be comparable to that of the HOS-based method.

Thus, we observe that the proposed HMM-based face finding scheme is capable of locating single as well as multiple faces in photographs. Though we have used the Euclidean norm for constructing the Δ_{ij}'s, we believe that the use of HOS-based difference measurement would further improve the performance of the HMM-based face finding scheme.

Table 1 gives a quantitative comparison of the various schemes based on the number of correct faces found and false matches on the test data base. For the scheme in [11], we have simply reproduced the reported results. It is mentioned therein that the performance of their scheme is marginally better than that of Poggio and Sung [6] over this data base. From the table, we note that the Euclidean-based system tends to perform rather poorly both in terms of missed faces and false alarms. The HOS-based scheme clearly outperforms the

TABLE 1
Performance Comparison of Rowley, Euclidean, HOS, and HMM Based Face Finding Systems

System	Missed faces/Total number of faces	(%) Face finding rate	False matches
Rowley	5/35	85.71	1
Euclidean	15/35	57.14	18
HOS	1/35	97.14	10
HMM	2/35	94.28	17

Euclidean-based face finder. This is mainly due to the refinement over the simple Gaussian fit, that the HOS-based method introduces. We note that the face finding rates of both the HOS-based and the HMM-based systems are very good. The false alarms are, however, slightly more and these are primarily nonface images. This may be attributed to the fact that our training set for nonface images was quite small (only 6364 patterns) as compared to 43,000 nonface patterns that are used in [6].

6. CONCLUSIONS

We have proposed two new schemes for finding faces in a cluttered image. In the first scheme, higher order statistics of the face and the face-like data samples are used to get a better approximation to the distributions of the face and the face-like manifolds. An HOS-based algorithm is used to perform data clustering. The algorithm is quite general in nature and would find potential applications in other pattern recognition tasks. The scheme successfully locates faces even in fairly complicated scenes. The second scheme that we have proposed here is based on the hidden Markov model. The basic idea is to learn face-to-nonface and nonface-to-face transitions in a photograph using an HMM. Once the HMM parameters corresponding to a given photograph are learned, the optimal state sequence decides the location of faces in the photograph. The HMM-based scheme also gives good results even on cluttered images. We mention here that the HOS-based method can be viewed as a supervised scheme, while the HMM-based method can be viewed as an unsupervised scheme for face finding in a cluttered environment. The overall performance of the proposed HOS-based and the HMM-based schemes is found to be quite good, even though, in comparison to existing popular schemes on face finding, our training set was very small.

We are currently exploring ways to speed up the HOS-based and the HMM-based methods for real-time applications. We are also working on the implementation of the HMM-based scheme with the difference measurements given by the HOS-based decision measure to further improve its performance.

APPENDIX A

Data Acquisition

To build a comprehensive but tractable data base of face patterns, we collected about 501 frontal views of human faces. Face images were captured locally using a camera setup in

0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

FIG. 6. A typical mask: 0's represent the cropped region.

the laboratory. Some were downloaded from image data bases and other image sources. The size of the face data base was further increased by adding mirror images and slightly rotated versions of the original face images. The images were then cropped to eliminate near-boundary pixels. A typical 17×17 mask for cropping is shown below in Fig. 6. Our face image data base thus consisted of 2004 masked canonical face patterns. To generate the face-like data base, we used the "boot-strapping" technique discussed in [6]. The face-like data base was incrementally generated by first building a reduced version of the system with our face clusters and a small set of nonface patterns. The false positive patterns that the system finds over a large set of natural images without faces are used to generate the data base of face-like patterns. The boot-strapping strategy reduces the number of nonface patterns needed to train a robust face finder. The face-like patterns were then cropped using the same mask that was used for the face patterns. Our face-like data base consisted of 4065 masked face-like patterns.

APPENDIX B

Morphological Operations in the Postprocessor of HMM-Based Scheme

It is assumed that the optimal state sequence, $\{\mathcal{S}^*_{i,j}\}_{i,j=1}^{M,N}$, has been binarized.

1. Erode the binary image using the following 2×2 template

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix};$$

"1" is the pixel under consideration which retains the value "1" if all the pixels shown by "0" in the 2×2 are also "1"s else the state value is set to "0." This template is used to erode the image from top to bottom and left to right. Then,

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

is used to erode the image from bottom to top and from right to left.

Carry out the erosion as described above for a predetermined number of iterations. The idea of using these two templates in two different directions is to find the representative

center of the cluster of "1"s rather than the "biased" cluster center. By this we mean that if we just used one of the templates and in that direction (say the first template traversing the image from top to bottom and left to right) then we would get an eroded image of the cluster which is in the right lower corner of the cluster of "1"s.

2. We use 2×1 templates to erode the image obtained from step 1. The templates used from top to bottom and right to left are

$$[0 \quad 1], \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 & \\ & 1 \end{bmatrix},$$

and the templates used to traverse the image from bottom to top and from right to left are

$$[1 \quad 0], \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \\ & 0 \end{bmatrix}.$$

Erosion is carried out for a predetermined number of iterations as described above.

APPENDIX C

Nomenclature

X	Random vector.
E	Expectation operator.
Φ	Moment generating function.
$f(x)$	Multivariate probability density function.
μ	Mean of X.
R	Covariance matrix of X.
$N(\cdot, \cdot)$	Gaussian probability density function.
$\otimes$	Tensor product.
H_k	Hermite polynomial of order k.
H_n	Vector whose elements are product of Hermite polynomials of order k_i, $i = 1 \ldots N$, such that $\sum_{i=1}^{N} k_i = n$.
π	Vector representing initial probability in HMM.
A	State transition probability matrix of HMM.
B	Matrix representing the probability of selecting a symbol while in a given state.
λ	HMM parameters $\{A, B, \pi\}$.
$\mathcal{N}$	Number of states of HMM.
$A_{i,j}$	Subimage at (i, j)th location generated by raster scanning.
$A'_{i,j}$	A_{ij} lexicographically ordered.
$\Delta_{i,j}$	Observation sequence of HMM.
$S_{i,j}$	State sequence of HMM.

ACKNOWLEDGMENTS

The authors thank the reviewers for their thoughtful comments. Thanks are also due to the various people who willingly came forward to pose for our face image collection system. The authors gratefully acknowledge the Olivetti Research Lab. and the MIT Media Lab. from where many face images were downloaded. Finally, the

authors thank Nilesh Hiremath for his enthusiastic support in setting up the photo studio and Mohammed Yeasin for several interesting discussions on the face finding problem.

REFERENCES

1. T. Sakai, M. Nagao, and S. Fujibayashi, Line extraction and pattern detection in a photograph, *Pattern Recognit.* **2**, 1969, 233–248.
2. M. A. Fischler and R. A. Elschlager, The representation and matching of pictorial structures, *IEEE Trans. Comput.* **22**, 1973, 67–92.
3. A. Yuille, D. Cohen, and P. Hallinan, *Facial Feature Extraction by Deformable Templates*, Technical Report 88-2, Harvard Robotics Laboratory, 1988.
4. G. Yang and T. S. Huang, Human face detection in a scene, in *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 1993*, pp. 453–458.
5. P. Sinha, Object recognition via image invariants: A case study, *Invest. Opthalmol. Visual Sci.* **35**, 1994, 1735–1740.
6. T. Poggio and K. Sung, Finding human faces with a Gaussian mixture distribution-based face model, in *Proc. Asian Conf. on Computer Vision, Singapore, 1995*, pp. 437–446.
7. K. Sung and T. Poggio, *Example-Based Learning for View-Based Human Face Detection, A.I. Memo No. 1521*, Massachusetts Institute of Technology, Dec. 1994.
8. B. Moghaddam and A. Pentland, Probabilistic visual learning for object detection, in *Proc. IEEE Intl. Conf. on Computer Vision, Cambridge, 1995*, pp. 786–793.
9. P. Seitz and G. K. Lang, Using local orientation and hierarchical spatial feature matching for the robust recognition of objects, in *Proceedings of SPIE, 1991*, pp. 252–259.
10. R. Valliant, C. Monrocq, and Y. L. Cun, Original approach for the location of objects in images, in *Proc. IEEE Intl. Conf. on Artificial Neural Networks*, 1993, pp. 26–29.
11. H. A. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection, in *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 1996*, pp. 203–208.
12. V. Govindaraju, Locating human faces in photographs, *Intl. J. Comput. Vision* **19**, 1996, 129–146.
13. A. N. Rajagopalan, K. Sunil Kumar, Jayashree Karlekar, R. Manivasakan, M. Milind Patil, U. B. Desai, P. G. Poonacha, and S. Chaudhuri, Finding faces in photographs, in *Proc. IEEE Intl. Conf. on Computer Vision, Bombay, India, 1998*, pp. 640–645.
14. H. Cramer, *Mathematical Methods of Statistics*, Princeton Univ. Press, Princeton, NJ, 1946.
15. G. M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 1, Charles Griffin, London, 1958.
16. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice–Hall, Englewood Cliffs, NJ, 1988.
17. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
18. F. Samaria and S. Young, HMM-based architecture for face identification, *Image Vision Comput.*, 1994, 537–543.
19. L. R. Rabiner and B. H. Juang, An introduction to hidden Markov models, *IEEE ASSP Magazine*, 1986, 4–16.
20. R. Dugad and U. B. Desai, *A Tutorial on Hidden Markov Models*, Technical Report No. SPANN-96.1, Indian Institute of Technology, Bombay, May 1996. [also at http://144.16.100.100/spann/thesis/Tech_Reports/HMM_TUT/hmm_tut.html]
21. G. D. Forney, Jr., The Viterbi algorithm, *Proc. IEEE* **61**, 1973, 263–278.