Finding Faces in Photographs

A. N. Rajagopalan, K. Sunil Kumar, Jayashree Karlekar, R. Manivasakan, M. Milind Patil,
U. B. Desai, P. G. Poonacha and S. Chaudhuri
Signal Processing and Artificial Neural Networks Laboratory
Department of Electrical Engineering
Indian Institute of Technology
Powai, Mumbai 400 076, India

Abstract

Two new schemes are presented for finding human faces in a photograph. The first scheme approximates the unknown distributions of the face and the face-like manifolds using higher order statistics (HOS). An HOS-based data clustering algorithm is also proposed. In the second scheme, the face to non-face and non-face to face transitions are learnt using a hidden Markov model (HMM). The HMM parameters are estimated corresponding to a given photograph and the faces are located by examining the optimal state sequence of the HMM. Experimental results are presented on the performance of both the schemes.

I. Introduction

The problem that we address in this paper is as follows: given a cluttered image, locate human faces in it. Face finding can serve as an important initial step towards building a fully automated face recognition system. It also finds applications in human-computer interfaces and surveillance/census systems. Related works on face-finding can be found in [1] - [8].

In this paper, we propose two new schemes for face finding. The first scheme is based on using higher order statistics (HOS) while the second is an unsupervised scheme based on the hidden Markov model (HMM). In the HOS-based scheme, our approach is distribution-based as in [3], but there are important differences. In [3], a simple Gaussian fit is used to model the distribution of the face and the face-like manifolds. We use higher order statistics of the face and the face-like data samples to get a better approximation to the unknown distribution. 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 pre-processing operations on the face and the face-like patterns unlike in [7]. The second approach that we propose here is based on the concept of learning the face to non-face and non-face to face transitions in a photograph. The approach is based on generating an observation sequence from the photograph and learning the HMM parameters corresponding to this sequence. The post-processed optimal state sequence decides the location of faces in the photograph. Both the HOS-based and the HMM-based schemes locate faces successfully even in images with multiple faces in a complex background.

II. 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 [9]. The HOS-based decision measure is derived from this expansion.

Let the random vector $\underline{X} = [X_1\ X_2\ \ldots X_N]^T$ and $\underline{X} \sim N(\underline{0}, I)$. If $\underline{t} = [t_1\ t_2\ \ldots t_N]^T$, then the moment generating function of $\underline{X}$ is given by $\Phi(\underline{t}) = E\left[\exp\left(\underline{t}^T\underline{X}\right)\right]$. Since these random variables are statistically independent, $\Phi(\underline{t}) = \exp\left(\frac{1}{2}\underline{t}^T\underline{t}\right)$. Therefore, $E\left[\exp\left(\underline{t}^T\underline{X} - \frac{1}{2}\underline{t}^T\underline{t}\right)\right] = 1$. Replacing $\underline{t}$ by $\underline{t} + \underline{s}$, we get $E\left[\exp\left(\underline{t}^T\underline{X} - \frac{1}{2}\underline{t}^T\underline{t}\right)\exp\left(\underline{s}^T\underline{X} - \frac{1}{2}\underline{s}^T\underline{s}\right)\right] = \exp\left(\underline{t}^T\underline{s}\right)$. Expanding this equation in Taylor series, we obtain $E\left[\sum_{m,n=0}^{\infty}(\underline{t}^{\otimes n})^T\frac{\overline{H}_n(\underline{X})}{n!}\frac{\overline{H}_m^T(\underline{X})}{m!}(\underline{s}^{\otimes m})\right] = \sum_{m,n=0}^{\infty}\frac{1}{n!}(\underline{t}^{\otimes n})^T I_{q_nq_m}(\underline{s}^{\otimes m})$, where $\otimes$ represents tensor product and $\underline{t}^{\otimes n} = \underbrace{\underline{t}\otimes\underline{t}\otimes\ldots\otimes\underline{t}}_{n\text{ times}}$. The matrix $I_{q_nq_m} = O_{q_nq_m}$ for $n \neq m$ and $I_{q_nq_m} = I_{q_n}$ for $n = m$, where $O_{q_nq_m}$ 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 vector $\overline{H}_n(\underline{x}) = \left[(\underline{D_t}^{\otimes n})\exp\left(\underline{t}^T\underline{x} - \frac{1}{2}\underline{t}^T\underline{t}\right)\right]_{\underline{t}=0}$ and $\underline{D_t} = \left[\frac{\partial}{\partial t_1}\ \frac{\partial}{\partial t_2}\cdots\frac{\partial}{\partial t_N}\right]^T$. The dimensions of $\overline{H_n}(\underline{x})$ and $\overline{H_m}(\underline{x})$ are given by q_n and q_m, respectively. By equating the coefficients of $\underline{t}$ and $\underline{s}$ on both sides, we obtain the important orthogonality relation

$$E\left[\underline{H}_n(\underline{X})\underline{H}_m^T(\underline{X})\right] = I_{p_np_m}\ . \qquad (1)$$

Note that the expectation is w.r.t $N(\underline{0}, I)$. In (1), $\underline{H}_n(\underline{x})$ 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 $\underline{H}_n(\underline{x})$ and $\underline{H}_m(\underline{x})$ 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 $\left[\frac{\partial^{k_i}}{\partial t_i^{k_i}} \exp\left(t_i x_i - \frac{1}{2}t_i^2\right)\right]_{t_i=0}$. Similarly, one can derive an orthogonality relation in terms of $N(\underline{\mu}, R)$ as [10]

$$E\left[\underline{H}_n\left(R^{-\frac{1}{2}}(\underline{X}-\underline{\mu})\right)\ \underline{H}_m^T\left(R^{-\frac{1}{2}}(\underline{X}-\underline{\mu})\right)\right] = I_{p_n p_m}.$$

Let $\underline{Y} = R^{-\frac{1}{2}}(\underline{X}-\underline{\mu})$ and $\underline{y} = R^{-\frac{1}{2}}(\underline{x}-\underline{\mu})$. If $\underline{X}$ has mean $\underline{\mu}$ and covariance R, then using the above orthogonality relation, the multivariate probability density function $f(\underline{x})$ can be shown to be [10]

$$f(\underline{x}) = N(\underline{\mu}, R)\left(1 + \sum_{n=3}^{\infty} E\left[\underline{H}_n^T(\underline{Y})\right]\underline{H}_n(\underline{y})\right). \quad (2)$$

III. 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 at all points of the image and across different scales.

A. Face Modeling

Analogous to [3], 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. The set of all 13 × 13 pixel canonical face patterns maps to a manifold in a multidimensional vector space. Since the global shape of this manifold may be quite complex and analytically intractable, one has to approximate the underlying distribution. In [3], the distribution of the face manifold was modeled by fitting the face data sample with multi-dimensional Gaussian clusters. It is our conjecture that the face pattern distribution is unlikely to be governed by a simple multi-dimensional Gaussian function. We approximate $f(\underline{x})$ by up to its m^{th} order joint moment by using (2) as

$$f(\underline{x}) = N(\underline{\mu}, R)\left(1 + \sum_{n=3}^{m} E\left[\underline{H}_n^T(\underline{Y})\right]\underline{H}_n(\underline{y})\right), \quad (3)$$

where $\underline{Y} = R^{-\frac{1}{2}}(\underline{X}-\underline{\mu})$ and $\underline{y} = R^{-\frac{1}{2}}(\underline{x}-\underline{\mu})$. Thus, higher order moments are used to get a better approximation to $f(\underline{x})$ than a simple Gaussian fit. For computational reasons, we choose $m = 3$. We too use (as in [3]) six clusters to describe the face distribution.

In the real world, many naturally occurring non-face 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. 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. The approach adopted for modeling the distribution of the face manifold is repeated for the face-like manifold too.

B. Data Clustering

When the actual distribution of the data is non-Gaussian, then traditional k-means algorithms based on Euclidean and Mahalanobis distances [11], [12] may fail to yield satisfactory results. Hence, we develop a clustering algorithm that uses higher order (> 2) statistics for improved clustering. We define the HOS-based decision measure as $(-\log f(\underline{x}))$. From (3) the HOS-based finite order decision measure is given by

$$-\log N(\underline{\mu}, R)\left(1 + \sum_{n=3}^{m} E\left[\underline{H}_n^T(\underline{Y})\right]\underline{H}_n(\underline{y})\right). \quad (4)$$

Interestingly, when $f(\underline{x})$ is Gaussian, the HOS-based decision measure neatly reduces to the normalized Mahalanobis distance.

B.1 HOS-based Clustering Algorithm

1. Obtain k initial pattern centers from the image database (either face or non-face). 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 re-assigning each data sample to the nearest pattern center using the HOS-based decision measure defined in (4). 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. Re-compute the moments (second and onwards 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 re-assigning 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.

C. Face Classification

A multi-layer perceptron net is used in our scheme for classification. The network has 12 input nodes, 6 hidden nodes and 1 output node. The input vector to the network consists of 12 elements that are difference measurements between the image pattern and the 12 (6 face and 6 face-like) clusters of the model. The differences are computed using the HOS-based measure given in (4). Once training is complete, to find whether a given image pattern is a face or not, a 12 dimensional vector of difference measurements corresponding to the test pattern is computed using (4). The trained classifier then determines whether or not the test pattern is a face.

IV. HMM-Based Face Finding

The face finding scheme using HMM can be broadly divided into three sections, *viz.* a pre-processing section, a processing section and a post-processing section. In the pre-processing section, the observation sequence that the HMM must learn is first generated 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 non-faces 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 trained on this sequence. After learning is complete, the post-processor first binarizes the optimal state sequence and then performs a 'cleaning' operation to remove the background clutter. The post-processed optimal state sequence gives the location of faces in the image. The HMM has been previously used, among others, by Samaria and Young for face recognition [13]. A detailed discussion on HMM can be found in [14], [15].

A. Pre-processing

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. Crop each subimage with an appropriate mask. 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 $\Delta_{i,j} = [\delta^1_{i,j}, \delta^2_{i,j}, \cdots, \delta^{12}_{i,j}]^T = [\, \|A'_{i,j} - K_1\|^2, \|A'_{i,j} - K_2\|^2, \cdots, \|A'_{i,j} - K_{12}\|^2 \,]^T$ using the face centroids $K_1, K_2, \cdots, K_6$ and the face-like centroids $K_7, K_8, \cdots, K_{12}$ from the knowledge-base.
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.

B. Processing

1. Determine the number of states $\mathcal{N}$ to be used for the HMM as follows:
   (a) Initialize $\eta = 2$ and η_{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, $[\, ^k\delta^1, \, ^k\delta^2, \cdots, \, ^k\delta^{12} \,]$).
   (d) for $k = 1$ to η, compute $\alpha_* = \min_{\alpha} {}^k\delta^{\alpha}$. If for some $k = k^*$, α_* is such that $1 \leq \alpha_* \leq 6$, then the bin corresponding to k^* represents a *face*. Note that δ's represent the distances from the knowledge-base (face$\{K_1, \cdots, K_6\}$, non-face$\{K_7, \cdots, K_{12}\}$). Output $\mathcal{N} = \eta$ as the number of states to be used by HMM, else $\eta = \eta + 1$, until η_{max}. If 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 $\{\mathcal{S}^0_{i,j}\}$ is next obtained using Viterbi algorithm [16].
   for iter = 1 to ITER (ITER predefined)
   (i) $\max_{\lambda} \mathcal{P}\,[\{\Delta_{i,j}\}, \{\mathcal{S}^{iter}_{i,j}\} \mid \lambda]$ to determine λ.
   (ii) using λ, determine the optimal state sequence $\{\mathcal{S}^{iter+1}\}$
   (iii) if $\{\mathcal{S}^{iter}_{i,j}\}$ and $\{\mathcal{S}^{iter+1}_{i,j}\}$ 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 $iter = iter + 1$.

C. Post-processing

1. Binarize the optimal state sequence $\{\mathcal{S}^*_{i,j}\}$ by setting the state corresponding to face to 1 and all the other states to 0. The state corresponding to face is to be processed and the rest are assumed to correspond to the background.

Fig. 1. Output results of the HOS-based face finding scheme.

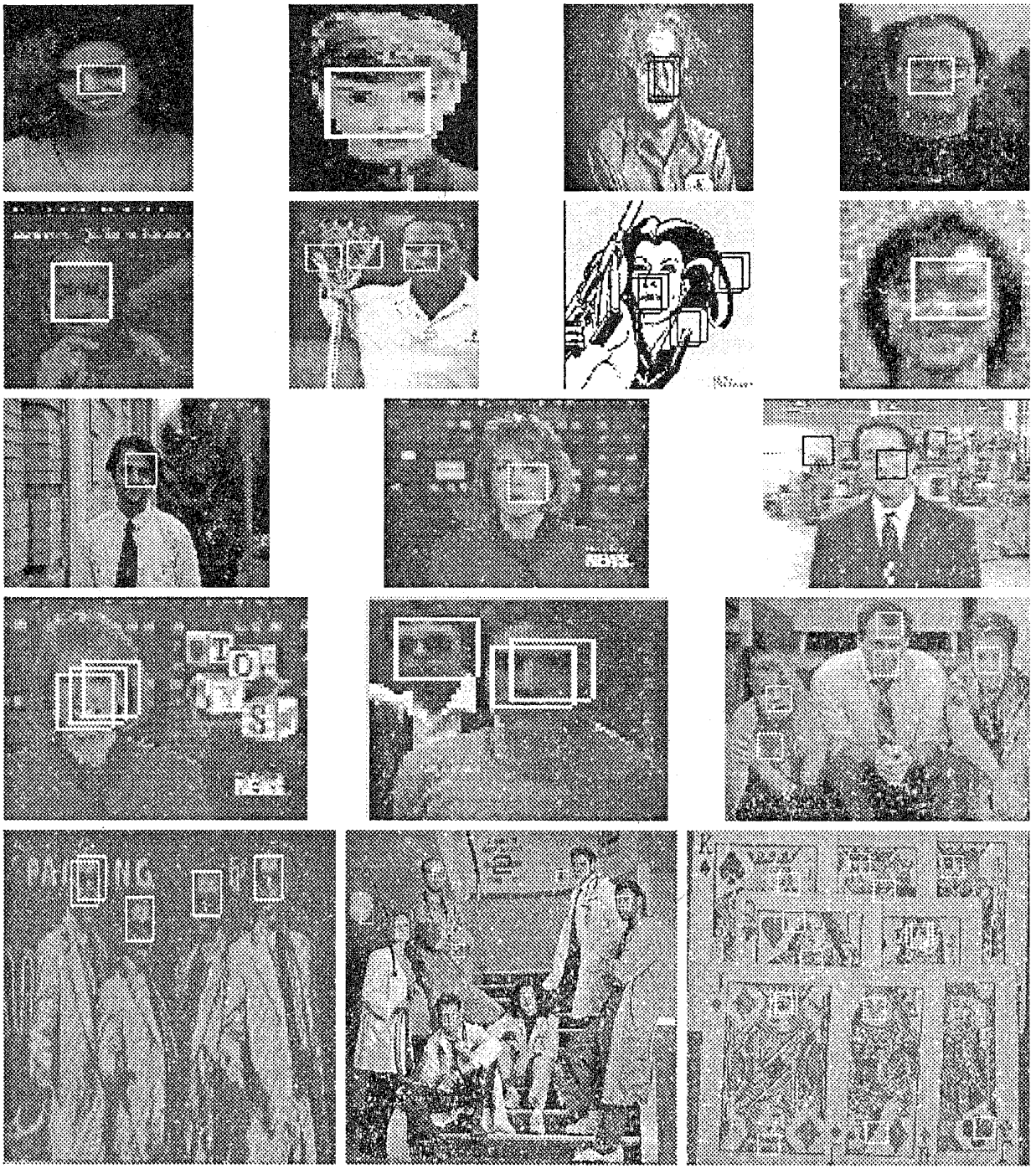

Fig. 2. Results of the HMM-based face finding scheme for the images in the test database.

2. Erode the binary image from top to bottom and left to right using the following 2×2 template $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. Here, "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". 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 for a predetermined number of iterations.
3. Use 2×1 templates to erode the image obtained from step 1 for a predetermined number of iterations. The templates used from top to bottom and right to left are $\begin{bmatrix} 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 \end{bmatrix}$ and the templates used to traverse the image from bottom to top and from right to left are $\begin{bmatrix} 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \end{bmatrix}$.

V. Experimental Results

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 [3] and [7]. The training set consisted of 2004 "face" patterns, 4065 "face-like" patterns and 6364 additional "non-face" patterns. It is important to note here that the size of our training set is much smaller than the one in [3], where 4150 face patterns, 6189 face-like patterns and about 43,000 non-face patterns are used to train the neural net. 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 used in [3] and 20×20 pixels in [7]. The test database reported in [7] was used for comparison of performance. This set is completely distinct from our training set. Due to space constraint, output results are given only for the HOS-based and the HMM-based face finding schemes. Nevertheless, for purpose of comparison, a quantitative breakdown of the performance of each of the above schemes is tabulated. Figs. 1 and 2 show the output results of the HOS-based and the HMM-based face finding systems, respectively. We observe that the proposed schemes are able to locate faces quite well in all the images in the test database. The systems work reliably, even for fairly complicated scenes.

Table 1 gives a quantitative comparison of the various schemes. For the scheme in [7], we have simply reproduced the reported results. 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 Euclidean-based face finder, as expected. Further, we note that both the HOS-based and the HMM-based systems have a better face finding rate than [7]. The false alarms are, however, slightly more and these are primarily non-face images. This may be attributed to the fact that our training set for non-face images was quite small.

Table 1 : Performance comparison of Rowley, Euclidean HOS and HMM-based face finding systems.

System	Missed faces	(%) Face finding rate	False matches
Rowley	5/35	85.71	1
Euc.	15/35	42.85	18
HOS	1/35	97.14	10
HMM	2/35	94.28	17

Acknowledgement

Thanks are 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.

References

[1] 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.

[2] P. Sinha, "Object recognition via image invariants: A case study", *Investigative Opthalmology and Visual Science*, vol. 35, pp. 1735-1740, 1994.

[3] T. Poggio and K. Sung, "Finding human faces with a Gaussian mixture distribution-based face model", in *Proc. Asian Conf. on Computer Vision*, (Singapore), Springer Verlag, Eds. S. Z. Li, D. P. Mittal, E. K. Teoh and H. Wan, 1995, pp. 437-446.

[4] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object detection", in *Proc. IEEE Intl. Conf. on Computer Vision*, (Cambridge), 1995, pp. 786-793.

[5] P. Seitz and G. K. Lang, "Using local orientation and hierarchical spatial feature matching for the robust recognition of objects", *Proceedings of SPIE*, pp. 252-259, 1991.

[6] 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.

[7] 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.

[8] V. Govindaraju, "Locating human faces in photographs", *Intl. Journal of Computer Vision*, vol. 19, no. 2, pp. 129-146, 1996.

[9] G. M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 1, Charles Griffin and Co. Ltd., 1958.

[10] A. N. Rajagopalan *et al.*, "Finding faces in a cluttered scene", Tech. Rep. No. SPANN-97.1, Indian Institute of Technology-Bombay, May 1997.

[11] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall Inc., Englewood Cliffs, 1988.

[12] R. O Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons Inc., 1973.

[13] F. Samaria and S. Young, "HMM-based architecture for face identification", *Image and Vision Computing*, pp. 537-543, 1994.

[14] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models", *IEEE ASSP Mgz.*, pp. 4-16, 1986.

[15] R. Dugad and U. B. Desai, "A tutorial on hidden Markov models", Tech. Rep. No. SPANN-96.1, Indian Institute of Technology-Bombay, May 1996.

[16] G. D. Forney Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, no. 3, pp. 263-278, 1973.