Probabilistic Visual Learning for Face Detection

Baback Moghaddam and Alex Pentland

Perceptual Computing Section, The Media Laboratory,
Massachusetts Institute of Technology
Room E15-383, 20 Ames Street, Cambridge MA 02139, U.S.A.
Telephone: (617) 253-3891, Fax:(617) 253-8874
Email: baback@media.mit.edu, sandy@media.mit.edu

Abstract

In this paper we present an unsupervised technique for visual learning which is based on density estimation in high-dimensional spaces using an eigenspace decomposition. This technique is a probabilistic extension of the "distance-from-feature-space" metric used for target detction in [26],[20]. An optimal density estimate is derived for the case of Gaussian data (used to model unimodal distributions). This probability density is then used to formulate a *maximum likelihood* estimation framework for visual search and detection for automatic object recognition. Our learning technique is applied to the probabilistic visual modeling and subsequent detection of human faces and facial features. We also describe a complete system for the automatic detection, normalization and recognition of human faces.

1 Object Detection

The standard detection paradigm in image processing is that of normalized correlation or template matching. However this approach is only optimal in the simplistic case of a *deterministic* signal embedded in additive white Gaussian noise. When we begin to consider a target *class* detection problem — *e.g,* finding a generic human face or a human hand in a scene — we must incorporate the underlying probability distribution of the object. Subspace methods and eigenspace decompositions are particularly well-suited to such a task since they provide a compact and *parametric* description of the object's appearance and also automatically identify the *degrees-of-freedom* of the underlying statistical variability.

In particular, the eigenspace formulation leads to a powerful alternative to standard detection techniques such as template matching or normalized correlation. The reconstruction error (or residual) of the eigenspace decomposition (referred to as the "distance-from-face-space" in the context of the work with "eigenfaces" [26]) is an effective indicator of similarity. The residual error is easily computed using the projection coefficients and the original signal energy. This detection strategy is equivalent to matching with a linear combination of *eigentemplates* and allows for a greater range of distortions in the input signal (including lighting, and moderate rotation and scale). In a statistical signal detection framework, the use of eigentemplates has been shown to yield superior performance in comparison with standard matched filtering [13][20].

Pentland *et el.* [20] used this formulation for a modular eigenspace representation of facial features where the corresponding residual — referred to as "distance-from-*feature*-space" or DFFS — was used for localization and detection. Given an input image, a saliency map was constructed by computing the DFFS at each pixel. When using M eigenvectors, this requires M convolutions (which can be efficiently computed using an FFT) plus an additional local energy computation. The global minimum of this distance map was then selected as the best estimate of the location of the target.

In this paper we will show that the DFFS can be interpreted as an estimate of a marginal component of the probability density of the object and that a complete estimate must also incorporate a second marginal density based on a complementary "distance-*in*-feature-space" (DIFS). Using our estimates of the object densities, we formulate the problem of target detection from the point of view of a maximum-likelihood (ML) estimation problem. Mainly, given the visual field, estimate the position (and scale) of the image region which is most representative of the target of interest. Computationally this is achieved by sliding an m-by-n observation window throughout the image and at each location computing the *likelihood* that the local subimage $\mathbf{x}$ is an instance of the target class Ω — *i.e,* $P(\mathbf{x}|\Omega)$. After this probability map is computed, we select the location corresponding to the highest likelihood as our ML estimate of the target location. Note that the likelihood map can be evaluated over the parameter space affecting the object's appearance which can include transformations such as scale and rotation.

2 Eigenspace Density Estimation

Our approach to automatic visual learning is based on density estimation. However, instead of applying estimation techniques directly to the original high-dimensional space of the imagery, we use an eigenspace decomposition to yield a computationally feasible estimate. Specifically, the eigenspace analysis is applied to a set of training views of the object in order to identify a principal subspace which captures the *intrinsic* dimensionality of the data. The component of the complete density in this lower-dimensional subspace is then estimated using a suitable parametric form. In addition, we implicitly model the component of the distribution in the *orthogonal* subspace. The complete density estimate can be efficiently computed from the lower-dimensional principal components. Our density estimate is shown to be *optimal* in the case of Gaussian-distributed training data.

Specifically, given a set of training images $\{\mathbf{x}^t\}_{t=1}^{N_T}$, from an object class Ω, we wish to estimate the class membership or *likelihood* function for this data — *i.e.,* $P(\mathbf{x}|\Omega)$. In this section, we examine two density estimation techniques for visual learning of high-dimensional data. Before introducing these estimators we briefly review eigenvector decomposition as commonly used in PCA.

2.1 Principal Component Imagery

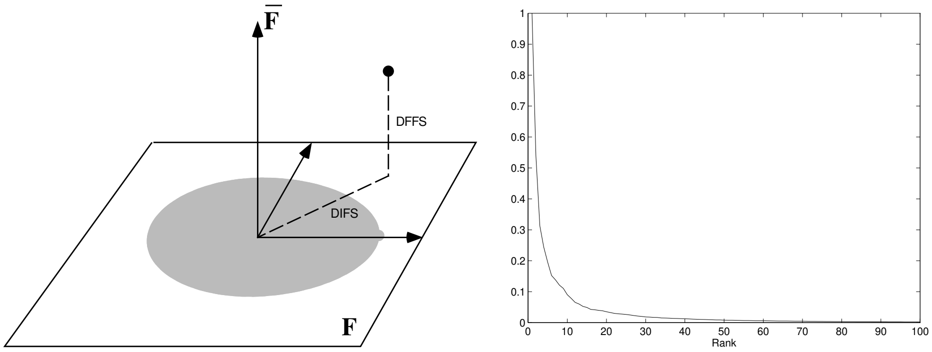
Given a training set of m-by-n images $\{I^t\}_{t=1}^{N_T}$, we can form a training set of vectors $\{\mathbf{x}^t\}$, where $\mathbf{x} \in \mathcal{R}^{N=mn}$, by lexicographic ordering of the pixel elements of each image I^t. The basis functions for the KLT [14] are obtained by solving the eigenvalue problem

$$\Lambda \;=\; \Phi^T \Sigma \Phi \tag{1}$$

where Σ is the covariance matrix, Φ is the eigenvector matrix of Σ and Λ is the corresponding diagonal matrix of eigenvalues. The unitary matrix Φ defines a coordinate transform (rotation) which *decorrelates* the data and makes explicit the *invariant subspaces* of the matrix operator Σ. In PCA, a partial KLT is performed to identify the largest-eigenvalue eigenvectors and obtain a principal component feature vector $\mathbf{y} = \Phi_{\mathrm{M}}^T\, \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ is the mean-normalized image vector and Φ_M is a submatrix of Φ containing the principal eigenvectors. PCA can be seen as a linear transformation $\mathbf{y} = \mathcal{T}(\mathbf{x}) : \mathcal{R}^N \rightarrow \mathcal{R}^M$ which extracts a lower-dimensional subspace of the KL basis corresponding to the maximal eigenvalues. These principal components preserve the major linear correlations in the data and discard the minor ones.[1]

By ranking the eigenvectors of the KL expansion with respect to their eigenvalues and selecting the first M principal components we form an orthogonal decomposition of the vector space $\mathcal{R}^N$ into two mutually exclusive and complementary subspaces: the principal subspace (or feature space) $F = \{\Phi_i\}_{i=1}^{M}$ containing the principal components and its orthogonal complement $\bar{F} = \{\Phi_i\}_{i=M+1}^{N}$. This orthogonal decomposition is illustrated in Figure 2.1(a) where we have a prototypical example of a distribution which is embedded entirely in F. In practice there is always a signal component in $\bar{F}$ due to the minor statistical variabilities in the data or simply due to the observation noise which affects every element of $\mathbf{x}$.

[1] In practice the number of training images N_T is far less than the dimensionality of the imagery N, consequently the covariance matrix Σ is singular. The first $M < N_T$ eigenvectors however can still be computed by working with the non-singular N_T-by-N_T *sample* covariance matrix Σ_s. The eigenvectors of Σ are linearly related to the eigenvectors of Σ_s [7].



(a) (b)

Figure 1: (a) Decomposition into the principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density (b) a typical eigenvalue spectrum obtained from KLT.

In a partial KL expansion, the residual reconstruction error is defined as

$$\epsilon^2(\mathbf{x}) \;=\; \sum_{i=M+1}^{N} y_i^2 \;=\; ||\tilde{\mathbf{x}}||^2 \;-\; \sum_{i=1}^{M} y_i^2 \tag{2}$$

and can be easily computed from the first M principal components and the L_2-norm of the mean-normalized image $\tilde{\mathbf{x}}$. Consequently the L_2 norm of every element $\mathbf{x} \in \mathcal{R}^N$ can be decomposed in terms of its projections in these two subspaces. We refer to the component in the orthogonal subspace $\bar{F}$ as the "distance-from-feature-space" (DFFS) which is a simple Euclidean distance and is equivalent to the residual error $\epsilon^2(\mathbf{x})$ in Eq.(2). The component of $\mathbf{x}$ which lies *in* the feature space F is referred to as the "distance-in-feature-space" (DIFS) but is generally not a distance-based norm, but can be interpreted in terms of the probability distribution of y in F.

2.2 Gaussian F-Space Densities

We begin by considering an optimal approach for estimating high-dimensional Gaussian densities. We assume that we have (robustly) estimated the mean $\bar{\mathbf{x}}$ and covariance Σ of the distribution from the given training set $\{\mathbf{x}^t\}$.[2] Under this assumption, the likelihood of a input pattern $\mathbf{x}$ is given by

$$P(\mathbf{x}|\Omega) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right]}{(2\pi)^{N/2}\,|\Sigma|^{1/2}} \tag{3}$$

The sufficient statistic for characterizing this likelihood is the *Mahalanobis* distance

$$d(\mathbf{x}) \;=\; \tilde{\mathbf{x}}^T\Sigma^{-1}\tilde{\mathbf{x}} \tag{4}$$

[2]In practice, a full rank N-dimensional covariance Σ can not be estimated from N_T independent observations where typically $N_T << N$. But as we shall see our estimator does not require the full covariance, but only its first M principal eigenvectors where $M < N_T << N$.

where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$. However, instead of evaluating this quadratic product explicitly, a much more efficient and robust computation can be performed, especially with regard to the matrix inverse Σ^{-1}. Using the eigenvectors and eigenvalues of Σ we can rewrite Σ^{-1} in the diagonalized form

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T \Sigma^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T \left[\Phi \Lambda^{-1} \Phi^T \right] \tilde{\mathbf{x}} \\ &= \mathbf{y}^T \Lambda^{-1} \mathbf{y} \end{aligned} \tag{5}$$

where $\mathbf{y} = \Phi^T \tilde{\mathbf{x}}$ are the new variables obtained by the change of coordinates in a KLT. Because of the diagonalized form, the *Mahalanobis* distance can also be expressed in terms of the sum

$$d(\mathbf{x}) = \sum_{i=1}^{N} \frac{y_i^2}{\lambda_i} \tag{6}$$

In the KLT basis, the *Mahalanobis* distance in Eq.(4) is conveniently *decoupled* into a weighted sum of uncorrelated component energies. Furthermore, the likelihood becomes a *product* of independent separable Gaussian densities. Despite its simpler form, evaluation of Eq.(6) is still computationally infeasible due to the high-dimensionality. We therefore seek to *estimate* $d(\mathbf{x})$ using only M projections. Intuitively, an obvious choice for a lower-dimensional representation is the principal subspace indicated by PCA which captures the major degrees of statistical variability in the data.[3] Therefore, we divide the summation into two independent parts corresponding to the principal subspace $F = \{\Phi_i\}_{i=1}^{M}$ and its orthogonal complement $\bar{F} = \{\Phi_i\}_{i=M+1}^{N}$

$$d(\mathbf{x}) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} + \sum_{i=M+1}^{N} \frac{y_i^2}{\lambda_i} \tag{7}$$

We note that the terms in the first summation can be computed by projecting $\mathbf{x}$ onto the M-dimensional principal subspace F. The remaining terms in the second sum $\{y_i\}_{i=M+1}^{N}$, however, can not be computed explicitly in practice because of the high-dimensionality. However, the *sum* of these terms is available and is in fact the DFFS quantity $\epsilon^2(\mathbf{x})$ which can be computed from Eq.(2). Therefore, based on the available terms, we can formulate an estimator for $d(\mathbf{x})$ as follows

$$\hat{d}(\mathbf{x}) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} + \frac{1}{\rho} \left[\sum_{i=M+1}^{N} y_i^2 \right] \tag{8}$$

$$= \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} + \frac{\epsilon^2(\mathbf{x})}{\rho} \tag{9}$$

where the term in the brackets is $\epsilon^2(\mathbf{x})$, which as we have seen can be computed using the first M principal components. We can therefore write the form of the likelihood estimate based on $\hat{d}(\mathbf{x})$ as the product of two marginal and independent Gaussian densities

$$\hat{P}(\mathbf{x}|\Omega) = \left[\frac{\exp\left(-\frac{1}{2} \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} \right)}{(2\pi)^{M/2} \prod_{i=1}^{M} \lambda_i^{1/2}} \right] \cdot \left[\frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho} \right)}{(2\pi\rho)^{(N-M)/2}} \right] = P_F(\mathbf{x}|\Omega)\ \hat{P}_{\bar{F}}(\mathbf{x}|\Omega) \tag{10}$$

[3] We will see shortly that given the typical eigenvalue spectra observed in practice (*e.g.,* Figure 2.1(b)), this choice is optimal for a different reason: it minimizes the information-theoretic *divergence* between the true density and our estimate of it.

where $P_F(\mathbf{x}|\Omega)$ is the true marginal density in F-space and $\hat{P}_{\bar{F}}(\mathbf{x}|\Omega)$ is the estimated marginal density in the orthogonal complement $\bar{F}$-space. The optimal value of ρ can now be determined by minimizing a suitable cost function $J(\rho)$. From an information-theoretic point of view, this cost function should be the Kullback-Leibler divergence or *relative entropy* [6] between the true density $P(\mathbf{x}|\Omega)$ and its estimate $\hat{P}(\mathbf{x}|\Omega)$

$$J(\rho) \;=\; \int P(\mathbf{x}|\Omega) \, \log \frac{P(\mathbf{x}|\Omega)}{\hat{P}(\mathbf{x}|\Omega)} \, d\mathbf{x} \;=\; \mathrm{E}\left[\log \frac{P(\mathbf{x}|\Omega)}{\hat{P}(\mathbf{x}|\Omega)}\right] \tag{11}$$

Using the diagonalized forms of the *Mahalanobis* distance $d(\mathbf{x})$ and its estimate $\hat{d}(\mathbf{x})$ and the fact that $\mathrm{E}[y_i^2] = \lambda_i$, it can be easily shown that

$$J(\rho) \;=\; \frac{1}{2} \sum_{i=M+1}^{N} \left[\frac{\lambda_i}{\rho} - 1 + \log \frac{\rho}{\lambda_i}\right] \tag{12}$$

The optimal weight ρ^* can be then found by minimizing this cost function with respect to ρ. Solving the equation $\frac{\partial J}{\partial \rho} = 0$ yields

$$\rho^* \;=\; \frac{1}{N-M} \sum_{i=M+1}^{N} \lambda_i \tag{13}$$

which is simply the arithmetic average of the eigenvalues in the orthogonal subspace $\bar{F}$. In addition to its optimality, ρ^* also results in an *unbiased* estimate of the *Mahalanobis* distance — *i.e,* $\mathrm{E}[\hat{d}(\mathbf{x};\rho^*)] = \mathrm{E}[d(\mathbf{x})]$. What this derivation shows is that once we select the M-dimensional principal subspace F (as indicated, for example, by PCA), the optimal estimate of the sufficient statistic $\hat{d}(\mathbf{x})$ has the form of Eq.(9) with ρ given by Eq.(13).

It is interesting to consider the minimal cost $J(\rho^*)$

$$J(\rho^*) \;=\; \frac{1}{2} \sum_{i=M+1}^{N} \log \frac{\rho^*}{\lambda_i} \tag{14}$$

from the point of view of the $\bar{F}$-space eigenvalues $\{\lambda_i : i = M+1, \cdots, N\}$. It is easy to show that $J(\rho^*)$ is minimized when the the $\bar{F}$-space eigenvalues have the *least* spread about their mean ρ^*. This suggests a strategy for selecting the principal subspace: choose F such that the eigenvalues associated with its orthogonal complement $\bar{F}$ have the least absolute deviation about their mean. In practice, the higher-order eigenvalues typically decay and stabilize near the observation noise variance. Therefore this strategy is usually consistent with the standard PCA practice of discarding the higher-order components since these tend to correspond to the "flattest" portion of the eigenvalue spectrum (see Figure 2.1(b)). In the limit, as the $\bar{F}$-space eigenvalues become exactly equal, the divergence $J(\rho^*)$ will be zero and our density estimate $\hat{P}(\mathbf{x}|\Omega)$ approaches the true density $P(\mathbf{x}|\Omega)$.

We note that in most applications it is customary to simply discard the $\bar{F}$-space component and simply work with $P_F(\mathbf{x}|\Omega)$. However, the use of the DFFS metric or equivalently the marginal density $P_{\bar{F}}(\mathbf{x}|\Omega)$ is critically important in formulating the likelihood of an observation $\mathbf{x}$ — especially in an object detection task — since there are an infinity of vectors which are *not* members of Ω which can have likely F-space projections. Without $P_{\bar{F}}(\mathbf{x}|\Omega)$ a detection system can result in a significant number of false alarms.

3 Maximum-Likelihood Detection

The density estimate $\hat{P}(\mathbf{x}|\Omega)$ can be used to compute a local measure of target saliency at each spatial position (i,j) in an input image based on the vector $\mathbf{x}$ obtained by the lexicographic ordering of the pixel values in a local neighborhood R

$$S(i,j;\Omega) = \hat{P}(\mathbf{x}|\Omega)\ , \quad \mathbf{x} \ = \downarrow [\{I(i+r,j+c) : (r,c) \in R\}] \tag{15}$$

where $\downarrow [\bullet]$ is the operator which converts a subimage into a vector. The ML estimate of position of the target Ω is then given by

$$(i,j)^{\mathrm{ML}} \ = \ \mathrm{argmax}\ S(i,j;\Omega) \tag{16}$$

This ML formulation can be extended to estimate object scale with *multiscale* saliency maps. The likelihood computation is performed (in parallel) on linearly scaled versions of the input image $I^{(\sigma)}$ corresponding to a pre-determined set of scales $\{\sigma_1, \sigma_2, \cdots \sigma_n\}$

$$S(i,j,k;\ \Omega) = \hat{P}\left(\downarrow \{I^{(\sigma_k)}(\sigma_k i + r, \sigma_k j + c) : (r,c) \in R\} \mid \Omega\right) \tag{17}$$

where the ML estimate of the spatial and scale indices is defined by

$$(i,j,k)^{\mathrm{ML}} \ = \ \mathrm{argmax}\ S(i,j,k\ ;\ \Omega) \tag{18}$$

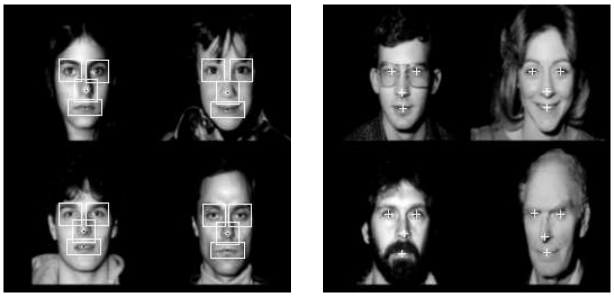
One important factor of variability in the appearance of the object in grayscale imagery is that of lighting and contrast. However, one can normalize for global illumination changes (as well as the linear response characteristics of the CCD camera) by normalizing each subimage $\mathbf{x}$ by its mean and standard deviation. This lighting normalization is performed both during training (density estimation) and also in the operational mode (*e.g.,* in detection).

4 Applications

The above ML detection technique has been tested in the detection of complex natural objects including human faces and facial features (*e.g.,* eyes). In this section we will present several examples from these application domains.

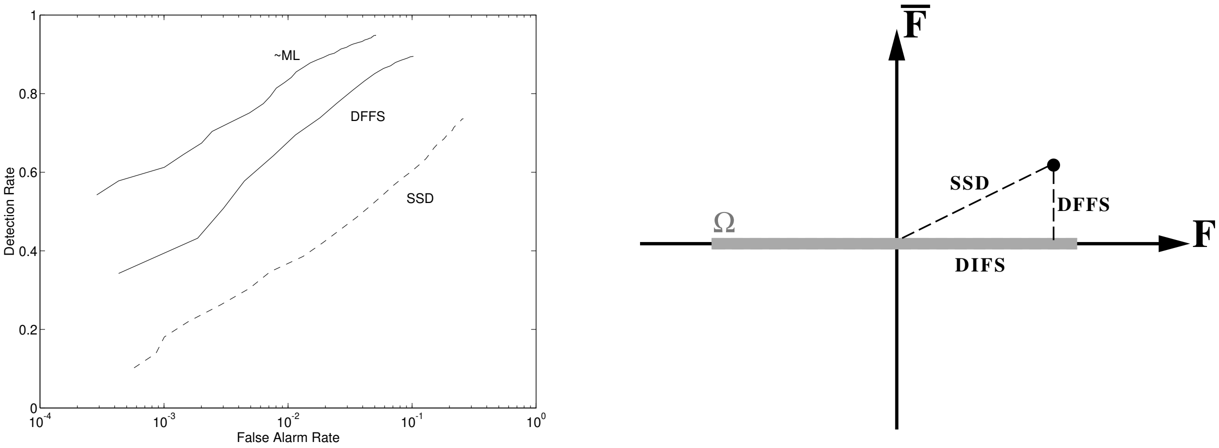
Over the years, various strategies for facial feature detection have been proposed, ranging from edge-map projections [10], to more recent techniques using generalized symmetry operators [24] and multilayer perceptrons [27]. In any robust face processing system this task is critically important since a face must be first geometrically normalized by aligning its features with those of a stored model before recognition can be attempted.

The eigentemplate approach to the detection of facial features in "mugshots" was proposed in [20], where the DFFS metric was shown to be superior to standard template matching for target detection. The detection task was the estimation of the position of facial features (the left and right eyes, the tip of the nose and the center of the mouth) in frontal view photographs of faces at fixed scale. Figure 2 shows examples of facial feature training templates and the resulting detections on the MIT Media Laboratory's database of 7,562 "mugshots".



(a) (b)

Figure 2: (a) Examples of facial feature training templates and (b) the resulting typical detections.



(a) (b)

Figure 3: (a) Detection performance of an SSD, DFFS and a ML detector, (b) geometric interpretation of the detectors.

We have compared the detection performance of three different detectors on approximately 7,000 test images from this database: a sum-of-square-differences (SSD) detector based on the average facial feature (in this case the left eye), an eigentemplate or DFFS detector and a ML detector based on $S(i,j;\Omega)$ as defined in section 2.2. Figure 3(a) shows the *receiver operating characteristic* (ROC) curves for these detectors, obtained by varying the detection threshold independently for each detector. The DFFS and ML detectors were computed based on a 5-dimensional principal subspace. Since the projection coefficients were unimodal a Gaussian distribution was used in modeling the true distribution for the ML detector as in section 2.2. Note that the ML detector exhibits the best detection vs. false-alarm tradeoff and yields the highest detection rate (of 95%). Indeed, at the *same* detection rate the ML detector has a false-alarm rate which is nearly 2 orders of magnitude lower than the SSD. Figure 3(b) provides the geometric intuition regarding the operation of these detectors. The SSD detector's threshold is based on the *radial* distance between the average

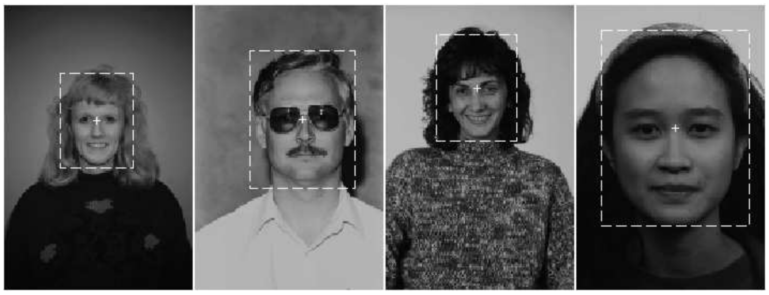


Figure 4: Multiscale Face Detection

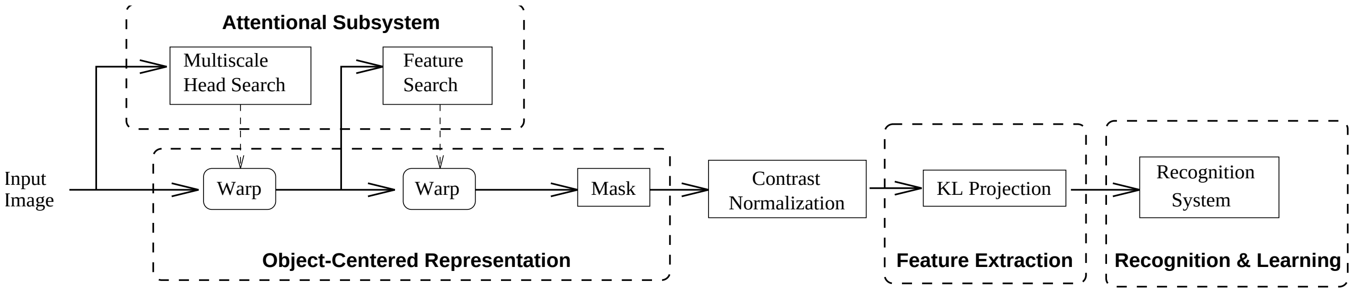


Figure 5: The face processing system.

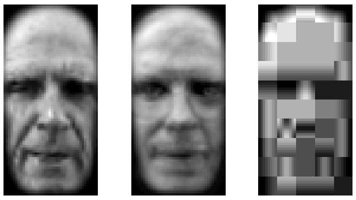
template (the origin of this space) and the input pattern. This leads to hyperspherical detection regions about the origin. In contrast, the DFFS detector measures the orthogonal distance to F, thus forming planar acceptance regions about F. Consequently to accept valid object patterns in Ω which are very different from the mean, the SSD detector must operate with high thresholds which consequently lead to many false alarms. But at the same time, the DFFS detector can not discriminate between the object class Ω and non-Ω patterns in F. The solution is provided by the ML detector which incorporates both the $\bar{F}$-space component (DFFS) and the F-space likelihood (DIFS). The probabilistic interpretation of Figure 3(b) is as follows: SSD assumes a *single* prototype (the mean) in additive white Gaussian noise whereas the DFFS assumes a *uniform* density in F. The ML detector, on the other hand, uses the complete probability density for detection.

We have incorporated and tested the multiscale version of the ML detection technique in a face detection task. This multiscale head finder was tested on the ARPA FERET database where 97% of 2,000 faces were correctly detected. Figure 4 shows examples of the ML estimate of the position and scale on these images. The multiscale saliency maps $S(i, j, k; \Omega)$ were computed based on the likelihood estimate $\hat{P}(\mathbf{x}|\Omega)$ in a 10-dimensional principal subspace using a Gaussian model (section 2.2). Note that this detector is able to localize the position and scale of the head despite variations in hair style and hair color, as well as presence of sunglasses. Illumination invariance was obtained by normalizing the input subimage $\mathbf{x}$ to a zero-mean unit-norm vector.

We have also used the multiscale version of the ML detector as the *attentional* component

(a) (b) (c) (d)

Figure 6: (a) original image, (b) position and scale estimate, (c) normalized head image, (d) position of facial features.



(a) (b) (c)

Figure 7: (a) aligned face, (b) eigenspace reconstruction (85 bytes) (c) JPEG reconstruction (530 bytes).

of an automatic system for recognition and model-based coding of faces. The block diagram of this system is shown in Figure 5 which consists of a two-stage object detection and alignment stage, a contrast normalization stage, and a feature extraction stage whose output is used for both recognition and coding. Figure 6 illustrates the operation of the detection and alignment stage on a natural test image containing a human face. The function of the face finder is to locate regions in the image which have a high likelihood of containing a face.

The first step in this process is illustrated in Figure 6(b) where the ML estimate of the position and scale of the face are indicated by the cross-hairs and bounding box. Once these regions have been identified, the estimated scale and position are used to normalize for translation and scale, yielding a standard "head-in-the-box" format image (Figure 6(c)). A second feature detection stage operates at this fixed scale to estimate the position of 4 facial features: the left and right eyes, the tip of the nose and the center of the mouth (Figure 6(d)). Once the facial features have been detected, the face image is warped to align the geometry and shape of the face with that of a canonical model. Then the facial region is extracted (by applying a fixed mask) and subsequently normalized for contrast. The geometrically aligned and normalized image (shown in Figure 7(a)) is then projected onto a custom set of eigenfaces to obtain a feature vector which is then used for recognition purposes as well as facial image coding. The software system for this face processing system is described in greater detail in the Appendix.

Figure 7 shows the normalized facial image extracted from Figure 6(d), its reconstruction using a 100-dimensional eigenspace representation (requiring only 85 bytes to encode) and a comparable

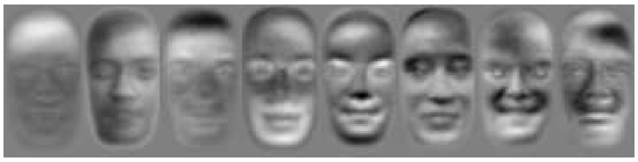


Figure 8: The first 8 eigenfaces.

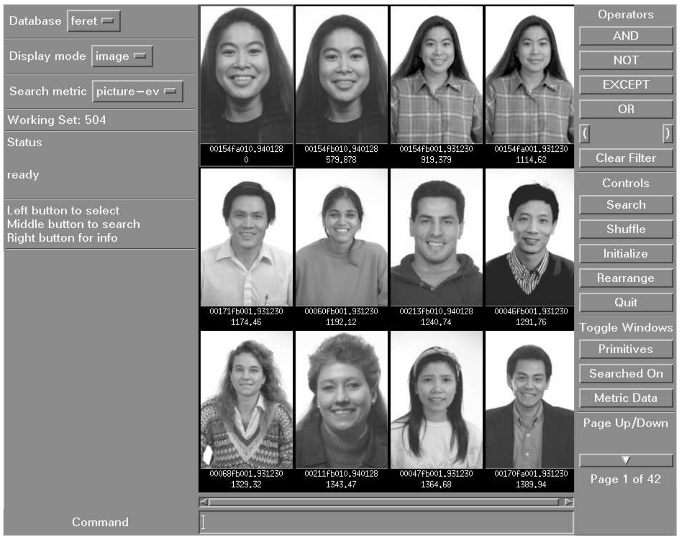


Figure 9: Photobook: FERET face database.

non-parametric reconstruction obtained using a standard transform-coding approach for image compression (requiring 530 bytes to encode). This example illustrates that the eigenface representation used for recognition is also an effective *model-based* representation for data compression. The first 8 eigenfaces used for this representation are shown in Figure 8.

Figure 9 shows the results of a similarity search in an image database tool called Photobook [21]. Each face in the database was automatically detected and aligned by the face processing system in Figure 5. The normalized faces were then projected onto a 100-dimensional eigenspace. The image in the upper left is the one searched on and the remainder are the ranked nearest neighbors in the FERET database. The top three matches in this case are images of the same person taken a month apart and at different scales. The recognition accuracy (defined as the percent correct rank-one matches) on a database of 155 individuals is 99% [15].

5 Discussion

In this paper we have described an eigenspace density estimation technique for unsupervised visual learning which exploits the *intrinsic* low-dimensionality of the training imagery to form a computationally simple estimator for the complete likelihood function of the object. Our estimator is based on a subspace decomposition and can be evaluated using only the M-dimensional principal component vector. We have derived the form for an optimal estimator and its associated expected cost for the case of a Gaussian density. In contrast to previous work on learning and characterization — which uses PCA primarily for dimensionality reduction and/or feature extraction — our method uses the eigenspace decomposition as an integral part of estimating *complete* density functions in high-dimensional image spaces. These density estimates were then used in a maximum-likelihood formulation for target detection. The multiscale version of this detection strategy was demonstrated in applications in which it functioned as an attentional subsystem for object recognition. The performance was found to be superior to existing detection techniques in experimental results on a large number of test data (on the order of thousands).

6 Appendix: Software

Here we discuss the specifics of the software functions employed in the face processing system. Below is a brief description of each stage in the pipeline and the associated functions and data files.

6.1 Head Detection

The function of this stage is to estimate the position and scale of the face and to rescale the input so as to obtain a head-centered image suitable for further processing. The detection of a face is performed using the function `ms_eigsearch` which performs the multiscale saliency computation as in section 3. Note that currently an *upper bound* on the likelihood is computed (using $\rho = \lambda_{M+1}$) instead of the optimal estimate (using ρ^*) as defined in section 2.2. Also in addition to the saliency computation based on likelihood, the current version of `ms_eigsearch` also incorporates likelihoods based on the first two moments of the grayscale histogram (mean and variance) as well as spatial location. Each of these two factors are incorporated independently using *Mahalanobis* distances based on previously computed means and covariances from training data. The mean and covariance for the spatial and grayscale prior probabilities are contained in the files `spatial_musigma.bf` and `umap_musigma.bf`, respectively. After the best head location and scale are determined the function `scalewarp` is used to linearly scale and translate the original image so as to center the head in the frame at a fixed (canonical) scale.

6.2 Feature Detection

Once a head-centered image is obtained the function `eigsearch` is used to conduct parallel searches for the four facial features: the eyes, the nose and the mouth. `eigsearch` is essentially identical to `ms_eigsearch` in its functionality, except that it computes likelihood maps at one scale and for several facial features. Like `ms_eigsearch`, `eigsearch` also currently implements a likelihood *upper bound* search (using $\rho = \lambda_{M+1}$) and carried out independently for each features. The saliency computation is restricted to certain regions (windows) in the head-centered frame and is also modulated by a prior probability distribution for the location of the features in these windows. This function returns the top N candidate locations (minima locations in the inverse likelihood map) for each feature. These minima locations are then input to the function `ml_detect` for verification and pruning of false alarms based on the geometrical constraints of a face (the relative locations of the individual features). This function performs an exhaustive combinatorial search of all possible N^4 pairings of the top N candidates for the 4 features. For each possible combination (which forms a candidate 4-node spatial graph), the function computes a likelihood score based on a *Mahalanobis* distance in terms of a 12-dimensional feature vector which consists of the length and orientation of the 6 links of this graph. The mean and covariance of this feature vector are contained in the file `geometry_musigma.bf`. In addition, the individual scores (likelihoods) of each candidate location is also taken into consideration. The final score is the product of these 4 individual likelihoods and the likelihood score from their geometry. `ml_detect` will return the feature locations corresponding to the best (most likely) spatial graph.

6.3 Facial Alignment

The final feature locations are then used to warp the head-centered image so as to align the detected feature locations with those of a canonical model. Two types of image warping are available: rigid (corresponding to a linear transform with no shearing) and affine (which is a general 6-parameter affine transform). Currently, a rigid transform is used based on the locations of the two eyes. The function `modelwarp` performs this warping operation based on the locations of the detected eyes and those of a canonical model (specified in the data file `model.bf`).

6.4 Facial Normalization

After this second stage of alignment, the warped image is masked (using the function `mask_image`), normalized for contrast (using the function `contrastmap`). The contrast normalization computes the mean and variance of the grayscale distribution and linearly remaps the grayscale to a specified mean (`pedestal`) and standard deviation (`scale`).

6.5 Recognition and Indexing

The normalized face image is then projected onto the set of eigenfaces to obtain a feature vector of projection coefficients (using the function `make_coeffs`). This feature vector is then used by the recognition module which consists of the function `pipe_neighbor` which computes the Euclidean distance between this vector and all others in the database and rank orders the entries.

7 Acknowledgements

The FERET database was provided by the US Army Research Laboratory.

References

[1] Anderson, C.H., Burt, P.J., and Van der Wall, G.S., "Change detection and tracking using pyramid transform techniques," *Proc. of SPIE Conf. on Intelligence, Robots and Computer Vision*, vol. 579, pp. 72-78, 1985.

[2] Bregler, C., and Omohundro, S.M., "Surface learning with applications to lip reading," in *Advances in Neural Information Processing Systems 6*, eds. J.D. Cowan, G. Tesauro and J. Alspector, Morgan Kaufman Publishers, San Fransisco, pp. 43-50, 1994.

[3] Brunelli, R., and Messelodi, S., "Robust estimation of correlation: an application to computer vision," *IRST Tech. Report no. 9310-015*, October 1993.

[4] Burl, M.C., *et al.*, "Automating the hunt for volcanos on Venus," *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, Seattle, WA, June 21-23, 1994.

[5] Cootes, T.F. and Taylor, C.J., "Active Shape Models: Smart Snakes," in *Proc. British Machine Vision Conf.*, Springer-Verlag, pp. 9-18, 1992.

[6] Cover, M. and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, New York, 1994.

[7] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[8] Hu, M.K., "Visual pattern recognition by moment invariants," *IEEE Trans. on Information Theory*, vol. 8, pp. 179-187, 1962.

[9] Jolliffe, I.T., *Principal Component Analysis*, Springer-Verlag, New York, 1986.

[10] Kanade, T., "Picture processing by computer complex and recognition of human faces," Tech. Report, Kyoto University, Dept. of Information Science, 1973.

[11] Kass, M., Witkin, A., Terzopoulos, D., "Snakes: active contour models," *ICCV'87*, pp. 259-268, 1987.

[12] Kirby, M., and Sirovich, L., "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, Jan. 1990.

[13] Kumar, B., Casasent, D., and Murakami, H., "Principal Component Imagery for Statistical Pattern Recognition Correlators," *Optical Engineering,* vol. 21, no. 1, Jan/Feb 1982.

[14] Loeve, M.M., *Probability Theory*, Van Nostrand, Princeton, 1955.

[15] Moghaddam, B. and Pentland, A., "Face recognition using view-based and modular eigenspaces," in *Automatic Systems for the Identification and Inspection of Humans*, SPIE vol. 2277. 1994.

[16] Murase, H., and Nayar, S. K., "Learning and Recognition of 3D Objects from Appearance" in *IEEE 2nd Qualitative Vision Workshop*, New York, NY, June 1993.

[17] Nayar, S.K., Murase, H.K., Nene, S.A, "General learning algorithm for robot vision," in *Neural & Stochastic Methods in Image & Signal Processing*, SPIE Proc. vol. 2304, July 1994.

[18] Palmer, S.E., "The psychology of perceptual organization: a transformational approach," in *Human and Machine Vision*, J. Beck, B. Hope and A. Rosenfeld (eds.), Academic Press, 1983.

[19] Pentland, A. and Sclaroff, S., "Closed form solutions for physically based shape modeling and recovery," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[20] Pentland, A., Moghaddam, B. and Starner, T., "View-based and modular eigenspaces for face recognition," *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, June 1994.

[21] Pentland, A., Picard, R., and Sclaroff, S., "Photobook: Tools for Content-Based Manipulation of Image Databases," SPIE Storage and Retrieval of Image and Video Databases II, No. 2185, San Jose, Feb 6-10, 1994.

[22] Persoon, E, and Fu, K.S., "Shape discrimination using Fourier descriptors," *Proc. 2nd IJCPR*, pp. 126-130, 1974.

[23] Redner, R.A., and Walker, H.F., "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195-239, 1984.

[24] Reisfeld, D., Wolfson, H., and Yeshurun, Y., "Detection of Interest Points Using Symmetry,", *ICCV '90*, Osaka, Japan, Dec. 1990.

[25] Sclaroff, S. and Pentland, A., "Modal Matching for Correspondence and Recognition," to appear: *IEEE Trans. on Pattern Analysis & Machine Intelligence.*

[26] Turk, M., and Pentland, A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* Vol. 3, No. 1, 1991.

[27] Vincent, J. M., Waite, J. B., and Myers, D. J, "Automatic Location of Visual Features by a System of Multilayered Perceptrons," *IEE Proceedings,* vol. 139, no. 6, Dec. 1992.

[28] Weng, J.J., "On comprehensive visual learning," in *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, Seattle, WA, June 24-25, 1994.