Homework 8: Eigenfaces for Recognition

Monday, November 2, 2020 11:23 PM

Eigenfaces-for-Recog...

Eigenfaces for Recognition

Matthew Turk and Alex Pentland
Vision and Modeling Group
The Media Laboratory
Massachusetts Institute of Technology

Abstract

■ We have developed a near-real-time computer system that can locate and track a subject's head, and then recognize the person by comparing characteristics of the face to those of known individuals. The computational approach taken in this system is motivated by both physiology and information theory, as well as by the practical requirements of near-real-time performance and accuracy. Our approach treats the face recognition problem as an intrinsically two-dimensional (2-D) recognition problem rather than requiring recovery of three-dimensional geometry, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. The system functions by projecting face images onto a feature space that spans the significant variations among known face images. The significant features are known as "eigenfaces," because they are the eigenvectors (principal components) of the set of faces; they do not necessarily correspond to features such as eyes, ears, and noses. The projection operation characterizes an individual face by a weighted sum of the eigenface features, and so to recognize a particular face it is necessary only to compare these weights to those of known individuals. Some particular advantages of our approach are that it provides for the ability to learn and later recognize new faces in an unsupervised manner, and that it is easy to implement using a neural network architecture. ■

INTRODUCTION

The face is our primary focus of attention in social intercourse, playing a major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable. We can recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expression, aging, and distractions such as glasses or changes in hairstyle or facial hair. As a consequence the visual processing of human faces has fascinated philosophers and scientists for centuries, including figures such as Aristotle and Darwin.

Computational models of face recognition, in particular, are interesting because they can contribute not only to theoretical insights but also to practical applications. Computers that recognize faces could be applied to a wide variety of problems, including criminal identification, security systems, image and film processing, and human–computer interaction. For example, the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to vastly improve criminal identification. Even the ability to merely detect faces, as opposed to recognizing them, can be important. Detecting faces in photographs, for instance, is an important problem in automating color film development, since the effect of many enhancement and noise reduction techniques depends on the picture content (e.g., faces should not be tinted green, while perhaps grass should).

Unfortunately, developing a computational model of face recognition is quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli. They are a natural class of objects, and stand in stark contrast to sine wave gratings, the "blocks world," and other artificial stimuli used in human and computer vision research (Davies, Ellis, & Shepherd, 1981). Thus unlike most early visual functions, for which we may construct detailed models of retinal or striate activity, face recognition is a very high level task for which computational approaches can currently only suggest broad constraints on the corresponding neural activity.

We therefore focused our research toward developing a sort of early, preattentive pattern recognition capability that does not depend on having three-dimensional information or detailed geometry. Our goal, which we believe we have reached, was to develop a computational model of face recognition that is fast, reasonably simple, and accurate in constrained environments such as an office or a household. In addition the approach is biologically implementable and is in concert with prelimi-

nary findings in the physiology and psychology of face recognition.

The scheme is based on an information theory approach that decomposes face images into a small set of characteristic feature images called "eigenfaces," which may be thought of as the principal components of the initial training set of face images. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces ("face space") and then classifying the face by comparing its position in face space with the positions of known individuals.

Automatically learning and later recognizing new faces is practical within this framework. Recognition under widely varying conditions is achieved by training on a limited number of characteristic views (e.g., a "straight on" view, a 45° view, and a profile view). The approach has advantages over other face recognition schemes in its speed and simplicity, learning capacity, and insensitivity to small or gradual changes in the face image.

Background and Related Work

Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, nose, mouth, and head outline, and defining a face model by the position, size, and relationships among these features. Such approaches have proven difficult to extend to multiple views, and have often been quite fragile, requiring a good initial guess to guide them. Research in human strategies of face recognition, moreover, has shown that individual features and their immediate relationships comprise an insufficient representation to account for the performance of adult human face identification (Carey & Diamond, 1977). Nonetheless, this approach to face recognition remains the most popular one in the computer vision literature.

Bledsoe (1966a,b) was the first to attempt semiautomated face recognition with a hybrid human–computer system that classified faces on the basis of fiducial marks entered on photographs by hand. Parameters for the classification were normalized distances and ratios among points such as eye corners, mouth corners, nose tip, and chin point. Later work at Bell Labs (Goldstein, Harmon, & Lesk, 1971; Harmon, 1971) developed a vector of up to 21 features, and recognized faces using standard pattern classification techniques. The chosen features were largely subjective evaluations (e.g., shade of hair, length of ears, lip thickness) made by human subjects, each of which would be quite difficult to automate.

An early paper by Fischler and Elschlager (1973) attempted to measure similar features automatically. They described a linear embedding algorithm that used local feature template matching and a global measure of fit to find and measure facial features. This template matching approach has been continued and improved by the recent work of Yuille, Cohen, and Hallinan (1989) (see Yuille, this volume). Their strategy is based on "deformable templates," which are parameterized models of the face and its features in which the parameter values are determined by interactions with the image.

Connectionist approaches to face identification seek to capture the configurational, or gestalt-like nature of the task. Kohonen (1989) and Kohonen and Lahtio (1981) describe an associative network with a simple learning algorithm that can recognize (classify) face images and recall a face image from an incomplete or noisy version input to the network. Fleming and Cottrell (1990) extend these ideas using nonlinear units, training the system by backpropagation. Stonham's WISARD system (1986) is a general-purpose pattern recognition device based on neural net principles. It has been applied with some success to binary face images, recognizing both identity and expression. Most connectionist systems dealing with faces (see also Midorikawa, 1988; O'Toole, Millward, & Anderson, 1988) treat the input image as a general 2-D pattern, and can make no explicit use of the configurational properties of a face. Moreover, some of these systems require an inordinate number of training examples to achieve a reasonable level of performance. Only very simple systems have been explored to date, and it is unclear how they will scale to larger problems.

Others have approached automated face recognition by characterizing a face by a set of geometric parameters and performing pattern recognition based on the parameters (e.g., Kaya & Kobayashi, 1972; Cannon, Jones, Campbell, & Morgan, 1986; Craw, Ellis, & Lishman, 1987; Wong, Law, & Tsaug, 1989). Kanade's (1973) face identification system was the first (and still one of the few) systems in which all steps of the recognition process were automated, using a top-down control strategy directed by a generic model of expected feature characteristics. His system calculated a set of facial parameters from a single face image and used a pattern classification technique to match the face from a known set, a purely statistical approach depending primarily on local histogram analysis and absolute gray-scale values.

Recent work by Burt (1988a,b) uses a "smart sensing" approach based on multiresolution template matching. This coarse-to-fine strategy uses a special-purpose computer built to calculate multiresolution pyramid images quickly, and has been demonstrated identifying people in near-real-time. This system works well under limited circumstances, but should suffer from the typical problems of correlation-based matching, including sensitivity to image size and noise. The face models are built by hand from face images.

THE EIGENFACE APPROACH

Much of the previous work on automated face recognition has ignored the issue of just what aspects of the face stimulus are important for identification. This suggested to us that an information theory approach of coding and

decoding face images may give insight into the information content of face images, emphasizing the significant local and global "features." Such features may or may not be directly related to our intuitive notion of face features such as the eyes, nose, lips, and hair. This may have important implications for the use of identification tools such as Identikit and Photofit (Bruce, 1988).

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.

In mathematical terms, we wish to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as a point (or vector) in a very high dimensional space. The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images.

These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that we can display the eigenvector as a sort of ghostly face which we call an *eigenface*. Some of the faces we studied are illustrated in Figure 1, and the corresponding eigenfaces are shown in Figure 2. Each eigenface deviates from uniform gray where some facial feature differs among the set of training faces; they are a sort of map of the variations between faces.

Each individual face can be represented exactly in terms of a linear combination of the eigenfaces. Each face can also be approximated using only the "best" eigenfaces—those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images. The best M eigenfaces span an M-dimensional subspace—"face space"—of all possible images.

The idea of using eigenfaces was motivated by a technique developed by Sirovich and Kirby (1987) and Kirby and Sirovich (1990) for efficiently representing pictures of faces using principal component analysis. Starting with an ensemble of original face images, they calculated a best coordinate system for image compression, where each coordinate is actually an image that they termed an *eigenpicture*. They argued that, at least in principle, any collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard pictures (the eigenpictures). The weights describing each face are found by projecting the face image onto each eigenpicture.

It occurred to us that if a multitude of face images can be reconstructed by weighted sums of a small collection of characteristic features or eigenpictures, perhaps an efficient way to learn and recognize faces would be to build up the characteristic features by experience over time and recognize particular faces by comparing the feature weights needed to (approximately) reconstruct them with the weights associated with known individuals. Each individual, therefore, would be characterized by the small set of feature or eigenpicture weights needed to describe and reconstruct them—an extremely compact representation when compared with the images themselves.

This approach to face recognition involves the following initialization operations:

1. Acquire an initial set of face images (the training set).
2. Calculate the eigenfaces from the training set, keeping only the M images that correspond to the highest eigenvalues. These M images define the *face space*. As new faces are experienced, the eigenfaces can be updated or recalculated.
3. Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their face images onto the "face space."

These operations can also be performed from time to time whenever there is free excess computational capacity.

Having initialized the system, the following steps are then used to recognize new face images:

1. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.
2. Determine if the image is a face at all (whether known or unknown) by checking to see if the image is sufficiently close to "face space."
3. If it is a face, classify the weight pattern as either a known person or as unknown.
4. (Optional) Update the eigenfaces and/or weight patterns.
5. (Optional) If the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into the known faces.

Calculating Eigenfaces

Let a face image $I(x,y)$ be a two-dimensional N by N array of (8-bit) intensity values. An image may also be considered as a vector of dimension N^2, so that a typical image of size 256 by 256 becomes a vector of dimension 65,536, or, equivalently, a point in 65,536-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal compo-

Figure 1. (a) Face images used as the training set.

nent analysis (or Karhunen–Loeve expansion) is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call "face space." Each vector is of length N^2, describes an N by N image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, we refer to them as "eigenfaces." Some examples of eigenfaces are shown in Figure 2.

Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_M$. The average face of the set is defined by $\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n$. Each face differs from the average by the vector $\Phi_i = \Gamma_i - \Psi$. An example training set is shown in Figure 1a, with the average face Ψ shown in Figure 1b. This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, $\mathbf{u}_n$, which best describes the distribution of the data. The kth vector, $\mathbf{u}_k$, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M} (\mathbf{u}_k^T \Phi_n)^2 \qquad (1)$$

is a maximum, subject to

$$\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

The vectors $\mathbf{u}_k$ and scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T \qquad (3)$$

$$= AA^T$$

where the matrix $A = [\Phi_1\ \Phi_2 \ldots \Phi_M]$. The matrix C, however, is N^2 by N^2, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space ($M < N^2$), there will be only $M - 1$, rather than N^2, meaningful eigenvectors. (The remaining eigenvectors will have associated eigenvalues of zero.) Fortunately we can solve for the N^2-dimensional eigenvectors in this case by first solving for the eigenvectors of an M by M matrix—e.g., solving a 16 × 16 matrix rather than a 16,384 × 16,384 matrix—

Figure 1. (b) The average face Ψ.

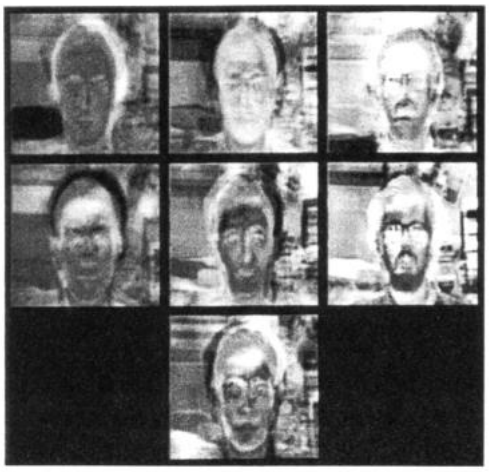

Figure 2. Seven of the eigenfaces calculated from the input images of Figure 1.

and then taking appropriate linear combinations of the face images $\mathbf{\Phi}_i$. Consider the eigenvectors $\mathbf{v}_i$ of A^TA such that

$$A^TA\mathbf{v}_i = \mu_i\mathbf{v}_i \tag{4}$$

Premultiplying both sides by A, we have

$$AA^TA\mathbf{v}_i = \mu_iA\mathbf{v}_i \tag{5}$$

from which we see that $A\mathbf{v}_i$ are the eigenvectors of $C = AA^T$.

Following this analysis, we construct the M by M matrix $L = A^TA$, where $L_{mn} = \mathbf{\Phi}_m^T\mathbf{\Phi}_n$, and find the M eigenvectors, $\mathbf{v}_l$, of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces $\mathbf{u}_l$.

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk}\mathbf{\Phi}_k, \qquad l = 1, \ldots, M \tag{6}$$

With this analysis the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the training set of face images will be relatively small ($M \ll N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images. Figure 2 shows the top seven eigenfaces derived from the input images of Figure 1.

Using Eigenfaces to Classify a Face Image

The eigenface images calculated from the eigenvectors of L span a basis set with which to describe face images. Sirovich and Kirby (1987) evaluated a limited version of this framework on an ensemble of $M = 115$ images of Caucasian males, digitized in a controlled manner, and found that about 40 eigenfaces were sufficient for a very good description of the set of face images. With $M' = 40$ eigenfaces, RMS pixel-by-pixel errors in representing cropped versions of face images were about 2%.

Since the eigenfaces seem adequate for describing face images under very controlled conditions, we decided to investigate their usefulness as a tool for face identification. In practice, a smaller M' is sufficient for identification, since accurate reconstruction of the image is not a requirement. In this framework, identification becomes a pattern recognition task. The eigenfaces span an M'-dimensional subspace of the original N^2 image space. The M' significant eigenvectors of the L matrix are chosen as those with the largest associated eigenvalues. In many of our test cases, based on $M = 16$ face images, $M' = 7$ eigenfaces were used.

A new face image ($\mathbf{\Gamma}$) is transformed into its eigenface components (projected into "face space") by a simple operation,

$$\omega_k = \mathbf{u}_k^T(\mathbf{\Gamma} - \mathbf{\Psi}) \tag{7}$$

for $k = 1, \ldots, M'$. This describes a set of point-by-point image multiplications and summations, operations performed at approximately frame rate on current image processing hardware. Figure 3 shows an image and its projection into the seven-dimensional face space.

The weights form a vector $\mathbf{\Omega}^T = [\omega_1, \omega_2 \ldots \omega_{M'}]$ that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The vector may then be used

in a standard pattern recognition algorithm to find which of a number of predefined face classes, if any, best describes the face. The simplest method for determining which face class provides the best description of an input face image is to find the face class k that minimizes the Euclidian distance

$$\vec{\epsilon}_k^2 = \|(\mathbf{\Omega} - \mathbf{\Omega}_k)\|^2 \quad (8)$$

where $\mathbf{\Omega}_k$ is a vector describing the kth face class. The face classes $\mathbf{\Omega}_i$ are calculated by averaging the results of the eigenface representation over a small number of face images (as few as one) of each individual. A face is classified as belonging to class k when the minimum ϵ_k is below some chosen threshold θ_ϵ. Otherwise the face is classified as "unknown," and optionally used to create a new face class.

Because creating the vector of weights is equivalent to projecting the original face image onto the low-dimensional face space, many images (most of them looking nothing like a face) will project onto a given pattern vector. This is not a problem for the system, however, since the distance ϵ between the image and the face space is simply the squared distance between the mean-adjusted input image $\mathbf{\Phi} = \mathbf{\Gamma} - \mathbf{\Psi}$ and $\mathbf{\Phi}_f = \sum_{i=1}^{M'} \omega_i \mathbf{u}_i$, its projection onto face space:

$$\epsilon^2 = \|\mathbf{\Phi} - \mathbf{\Phi}_f\|^2 \quad (9)$$

Thus there are four possibilities for an input image and its pattern vector: (1) near face space and near a face class, (2) near face space but not near a known face class, (3) distant from face space and near a face class, and (4) distant from face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is present. The last two cases indicate that the image is not a face image. Case three typically shows up as a false positive in most recognition systems; in our framework, however, the false recognition may be detected because of the significant distance between the image and the subspace of expected face images. Figure 4 shows some images and their projections into face space and gives a measure of distance from the face space for each.

Summary of Eigenface Recognition Procedure

To summarize, the eigenfaces approach to face recognition involves the following steps:

1. Collect a set of characteristic face images of the known individuals. This set should include a number of images for each person, with some variation in expression and in the lighting. (Say four images of ten people, so $M = 40$.)
2. Calculate the (40×40) matrix L, find its eigenvectors and eigenvalues, and choose the M' eigenvectors with the highest associated eigenvalues. (Let $M' = 10$ in this example.)
3. Combine the normalized training set of images according to Eq. (6) to produce the $(M' = 10)$ eigenfaces $\mathbf{u}_k$.
4. For each known individual, calculate the class vector $\mathbf{\Omega}_k$ by averaging the eigenface pattern vectors $\mathbf{\Omega}$ [from Eq. (8)] calculated from the original (four) images of the individual. Choose a threshold θ_ϵ that defines the maximum allowable distance from any face class, and a threshold θ_ϵ that defines the maximum allowable distance from face space [according to Eq. (9)].
5. For each new face image to be identified, calculate its pattern vector $\mathbf{\Omega}$, the distances ϵ_i to each known class, and the distance ϵ to face space. If the minimum distance $\epsilon_k < \theta_\epsilon$ and the distance $\epsilon < \theta_\epsilon$, classify the input face as the individual associated with class vector $\mathbf{\Omega}_k$. If the minimum distance $\epsilon_k > \theta_\epsilon$ but distance $\epsilon < \theta_\epsilon$, then the image may be classifed as "unknown," and optionally used to begin a new face class.
6. If the new image is classified as a known individual, this image may be added to the original set of familiar face images, and the eigenfaces may be recalculated (steps 1–4). This gives the opportunity to modify the face space as the system encounters more instances of known faces.

In our current system calculation of the eigenfaces is done offline as part of the training. The recognition currently takes about 400 msec running rather inefficiently in Lisp on a Sun4, using face images of size 128×128. With some special-purpose hardware, the current version could run at close to frame rate (33 msec).

Designing a practical system for face recognition within this framework requires assessing the tradeoffs between generality, required accuracy, and speed. If the face recognition task is restricted to a small set of people (such as the members of a family or a small company), a small set of eigenfaces is adequate to span the faces of interest. If the system is to learn new faces or represent many people, a larger basis set of eigenfaces will be required. The results of Sirovich and Kirby (1987) and Kirby and Sirovich (1990) for coding of face images gives some evidence that even if it were necessary to represent a large segment of the population, the number of eigenfaces needed would still be relatively small.

Locating and Detecting Faces

The analysis in the preceding sections assumes we have a centered face image, the same size as the training images and the eigenfaces. We need some way, then, to locate a face in a scene to do the recognition. We have developed two schemes to locate and/or track faces, using motion detection and manipulation of the images in "face space".

Figure 3. An original face image and its projection onto the face space defined by the eigenfaces of Figure 2.

Motion Detecting and Head Tracking

People are constantly moving. Even while sitting, we fidget and adjust our body position, nod our heads, look around, and such. In the case of a single person moving in a static environment, a simple motion detection and tracking algorithm, depicted in Figure 5, will locate and track the position of the head. Simple spatiotemporal filtering (e.g., frame differencing) accentuates image locations that change with time, so a moving person "lights up" in the filtered image. If the image "lights up" at all, motion is detected and the presence of a person is postulated.

After thresholding the filtered image to produce a binary motion image, we analyze the "motion blobs" over time to decide if the motion is caused by a person moving and to determine head position. A few simple rules are applied, such as "the head is the small upper blob above a larger blob (the body)," and "head motion must be reasonably slow and contiguous" (heads are not expected to jump around the image erratically). Figure 6 shows an image with the head located, along with the path of the head in the preceding sequence of frames.

The motion image also allows for an estimate of scale. The size of the blob that is assumed to be the moving head determines the size of the subimage to send to the recognition stage. This subimage is rescaled to fit the dimensions of the eigenfaces.

Using "Face Space" to Locate the Face

We can also use knowledge of the face space to locate faces in single images, either as an alternative to locating faces from motion (e.g., if there is too little motion or many moving objects) or as a method of achieving more precision than is possible by use of motion tracking alone. This method allows us to recognize the presence of faces apart from the task of identifying them.

As seen in Figure 4, images of faces do not change radically when projected into the face space, while the projection of nonface images appears quite different. This basic idea is used to detect the presence of faces in a scene: at every location in the image, calculate the distance ϵ between the local subimage and face space. This distance from face space is used as a measure of "faceness," so the result of calculating the distance from face space at every point in the image is a "face map" $\epsilon(x,y)$. Figure 7 shows an image and its face map—low values (the dark area) indicate the presence of a face.

Unfortunately, direct application of Eq. (9) is rather expensive. We have therefore developed a simpler, more efficient method of calculating the face map $\epsilon(x,y)$, which is described as follows.

To calculate the face map at every pixel of an image $I(x,y)$, we need to project the subimage centered at that pixel onto face space, then subtract the projection from the original. To project a subimage $\mathbf{\Gamma}$ onto face space, we must first subtract the mean image, resulting in $\mathbf{\Phi} = \mathbf{\Gamma} - \mathbf{\Psi}$. With $\mathbf{\Phi}_f$ being the projection of $\mathbf{\Phi}$ onto face space, the distance measure at a given image location is then

$$\begin{aligned}\epsilon^2 &= \|\mathbf{\Phi} - \mathbf{\Phi}_f\|^2 \\ &= (\mathbf{\Phi} - \mathbf{\Phi}_f)^T(\mathbf{\Phi} - \mathbf{\Phi}_f) \\ &= \mathbf{\Phi}^T\mathbf{\Phi} - \mathbf{\Phi}^T\mathbf{\Phi}_f + \mathbf{\Phi}_f^T(\mathbf{\Phi} - \mathbf{\Phi}_f) \\ &= \mathbf{\Phi}^T\mathbf{\Phi} - \mathbf{\Phi}_f^T\mathbf{\Phi}_f\end{aligned} \tag{10}$$

Note to the reader:

There are typos here that have haunted me for years, so here is a brief correction. First, the third line in equation (10) should not have a plus sign - rather, the plus should be replaced by a negative. Also, the second term of the fourth line should be $\Phi^T\Phi_f$ rather than $\Phi_f^T\Phi_f$.

This carries into equation (11). However, since the last term in the third line of equation (10) is equal to zero (due to the fact they're perpendicular), this means that these two terms are actually equivalent - i.e.,

$$\Phi^T\Phi_f = \Phi_f^T\Phi_f$$

So even though the wrong terms are written in the derivation, it's actually still correct.

Matthew Turk

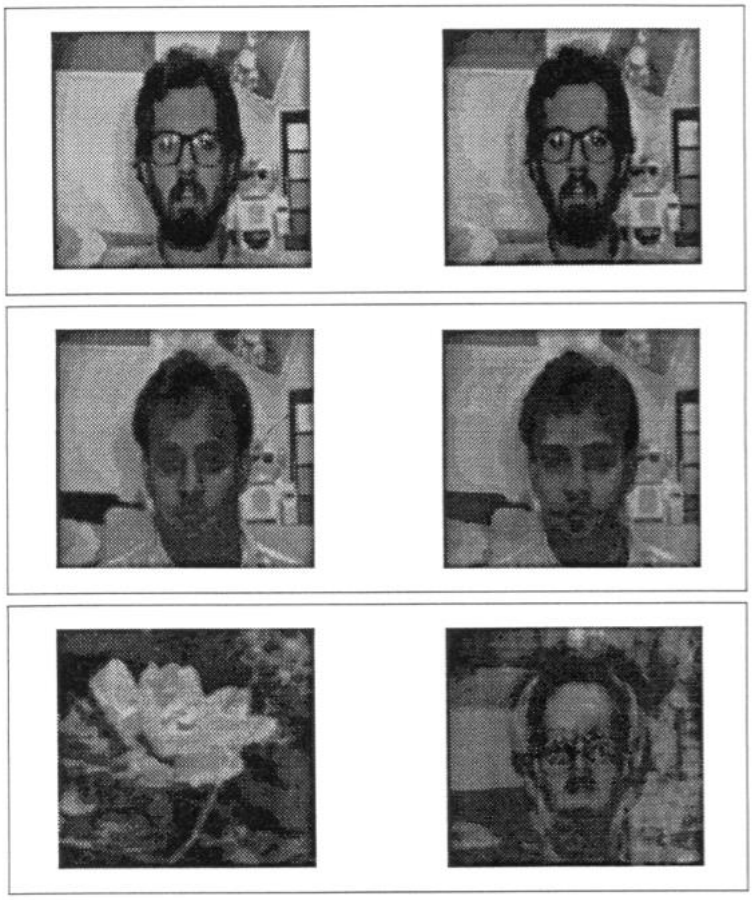

Figure 4. Three images and their projections onto the face space defined by the eigenfaces of Figure 2. The relative measures of distance from face space are **(a)** 29.8, **(b)** 58.5, **(c)** 5217.4. Images **(a)** and **(b)** are in the original training set.

since $\Phi_f \perp (\Phi - \Phi_f)$. Because Φ_f is a linear combination of the eigenfaces ($\Phi_f = \sum_{i=1}^{L} \omega_i \mathbf{u}_i$) and the eigenfaces are orthonormal vectors,

$$\Phi_f^T \Phi_f = \sum_{i=1}^{L} \omega_i^2 \tag{11}$$

and

$$\epsilon^2(x, y) = \Phi^T(x, y)\,\Phi(x, y) - \sum_{i=1}^{L} \omega_i^2(x, y) \tag{12}$$

where $\epsilon(x, y)$ and $\omega_i(x, y)$ are scalar functions of image location, and $\Phi(x, y)$ is a vector function of image location.

The second term of Eq. (12) is calculated in practice by a correlation with the L eigenfaces:

$$\begin{aligned}\sum_{i=1}^{L} \omega_i^2(x, y) &= \sum_{i=1}^{L} \Phi^T(x, y)\mathbf{u}_i \\ &= \sum_{i=1}^{L} [\Gamma(x, y) - \Psi]^T \mathbf{u}_i \\ &= \sum_{i=1}^{L} [\Gamma^T(x, y)\mathbf{u}_i - \Psi^T \mathbf{u}_i] \\ &= \sum_{i=1}^{L} [I(x, y) \otimes \mathbf{u}_i - \Psi^T \mathbf{u}_i]\end{aligned} \tag{13}$$

where $\otimes$ is the correlation operator. The first term of Eq. (12) becomes

$$\begin{aligned}\Phi^T(x, y)\Phi(x, y) &= [\Gamma(x, y) - \Psi]^T[\Gamma(x, y) - \Psi] \\ &= \Gamma^T(x, y)\Gamma(x, y) - 2\Psi^T\Gamma(x, y) + \Psi^T\Psi \\ &= \Gamma^T(x, y)\Gamma(x, y) - 2\Gamma(x, y) \otimes \Psi + \\ &\quad \Psi^T\Psi\end{aligned} \tag{14}$$

so that

$$\epsilon^2(x, y) = \Gamma^T(x, y)\Gamma(x, y) - 2\Gamma(x, y) \otimes \Psi + \Psi^T\Psi + \sum_{i=1}^{L} [\Gamma(x, y) \otimes \mathbf{u}_i - \Psi \otimes \mathbf{u}_i] \tag{15}$$

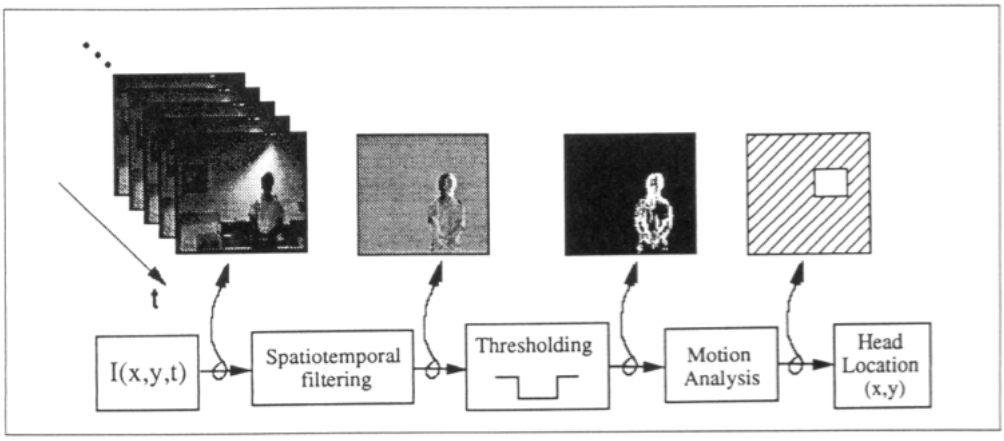

Figure 5. The head tracking and locating system.

Figure 6. The head has been located—the image in the box is sent to the face recognition process. Also shown is the path of the head tracked over several previous frames.

Since the average face Ψ and the eigenfaces $\mathbf{u}_l$ are fixed, the terms $\Psi^T\Psi$ and $\Psi \otimes \mathbf{u}_l$ may be computed ahead of time.

Thus the computation of the face map involves only $L + 1$ correlations over the input image and the computation of the first term $\Gamma^T(x, y)\Gamma(x, y)$. This is computed by squaring the input image $I(x, y)$ and, at each image location, summing the squared values of the local subimage. As discussed in the section on Neural Networks, these computations can be implemented by a simple neural network.

Learning to Recognize New Faces

The concept of face space allows the ability to learn and subsequently recognize new faces in an unsupervised manner. When an image is sufficiently close to face space but is not classified as one of the familiar faces, it is initially labeled as "unknown." The computer stores the pattern vector and the corresponding unknown image. If a collection of "unknown" pattern vectors cluster in the pattern space, the presence of a new but unidentified face is postulated.

The images corresponding to the pattern vectors in the cluster are then checked for similarity by requiring that the distance from each image to the mean of the images is less than a predefined threshold. If the images pass the similarity test, the average of the feature vectors is added to the database of known faces. Occasionally, the eigenfaces may be recalculated using these stored images as part of the new training set.

Other Issues

A number of other issues must be addressed to obtain a robust working system. In this section we will briefly mention these issues and indicate methods of solution.

Eliminating the Background

In the preceding analysis we have ignored the effect of the background. In practice, the background can significantly effect the recognition performance, since the ei-

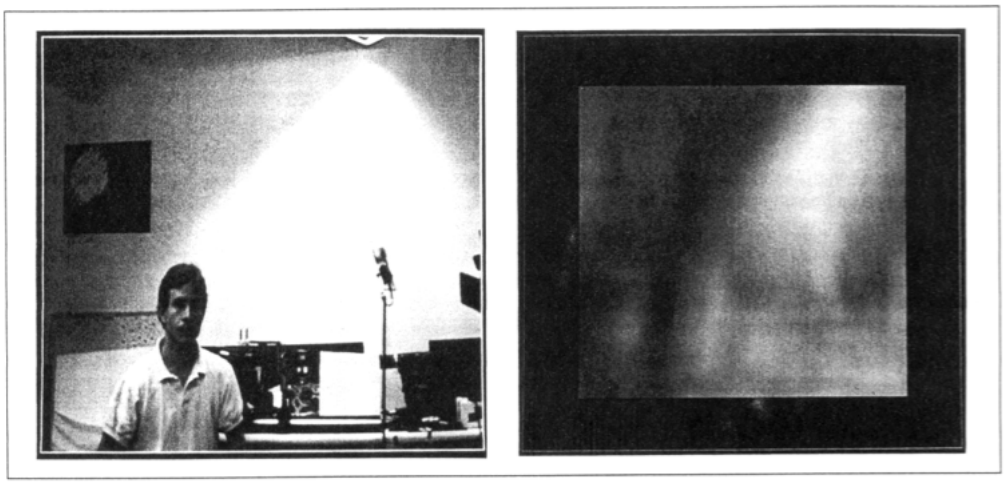

Figure 7. **(a)** Original image. **(b)** The corresponding face map, where low values (dark areas) indicate the presence of a face.

genface analysis as described above does not distinguish the face from the rest of the image. In the experiments described in the section on Experiments with Eigenfaces, the background was a significant part of the image used to classify the faces.

To deal with this problem without having to solve other difficult vision problems (such as robust segmentation of the head), we have multiplied the input face image by a two-dimensional gaussian window centered on the face, thus diminishing the background and accentuating the middle of the face. Experiments in human strategies of face recognition (Hay & Young, 1982) cite the importance of the internal facial features for recognition of familiar faces. Deemphasizing the outside of the face is also a practical consideration since changing hairstyles may otherwise negatively affect the recognition.

Scale (Head Size) and Orientation Invariance

The experiments in the section on Database of Face Images show that recognition performance decreases quickly as the head size, or scale, is misjudged. The head size in the input image must be close to that of the eigenfaces for the system to work well. The motion analysis gives an estimate of head size, from which the face image is rescaled to the eigenface size.

Another approach to the scale problem, which may be separate from or in addition to the motion estimate, is to use multiscale eigenfaces, in which an input face image is compared with eigenfaces at a number of scales. In this case the image will appear to be near the face space of only the closest scale eigenfaces. Equivalently, we can scale the input image to multiple sizes and use the scale that results in the smallest distance measure to face space.

Although the eigenfaces approach is not extremely sensitive to head orientation (i.e., sideways tilt of the head), a non-upright view will cause some performance degradation. An accurate estimate of the head tilt will certainly benefit the recognition. Again, two simple methods have been considered and tested. The first is to calculate the orientation of the motion blob of the head. This is less reliable as the shape tends toward a circle, however. Using the fact that faces are reasonably symmetric patterns, at least for frontal views, we have used simple symmetry operators to estimate head orientation. Once the orientation is estimated, the image can be rotated to align the head with the eigenfaces.

Distribution in Face Space

The nearest-neighbor classification previously described assumes a Gaussian distribution in face space of an individual's feature vectors Ω. Since there is no a priori reason to assume any particular distribution, we want to characterize it rather than assume it is gaussian. Nonlinear networks such as described in Fleming and Cottrell (1990) seem to be a promising way to learn the face space distributions by example.

Multiple Views

We are currently extending the system to deal with other than full frontal views by defining a limited number of face classes for each known person corresponding to *characteristic views*. For example, an individual may be represented by face classes corresponding to a frontal

face view, side views, at ± 45°, and right and left profile views. Under most viewing conditions these seem to be sufficient to recognize a face anywhere from frontal to profile view, because the real view can be approximated by interpolation among the fixed views.

EXPERIMENTS WITH EIGENFACES

To assess the viability of this approach to face recognition, we have performed experiments with stored face images and built a system to locate and recognize faces in a dynamic environment. We first created a large database of face images collected under a wide range of imaging conditions. Using this database we have conducted several experiments to assess the performance under known variations of lighting, scale, and orientation. The results of these experiments and early experience with the near-real-time system are reported in this section.

Database of Face Images

The images from Figure 1a were taken from a database of over 2500 face images digitized under controlled conditions. Sixteen subjects were digitized at all combinations of three head orientations, three head sizes or scales, and three lighting conditions. A six level Gaussian pyramid was constructed for each image, resulting in image resolution from 512 × 512 pixels down to 16 × 16 pixels. Figure 8 shows the images from one pyramid level for one individual.

In the first experiment the effects of varying lighting, size, and head orientation were investigated using the complete database of 2500 images of the 16 individuals shown in Figure 1a. Various groups of 16 images were selected and used as the training set. Within each training set there was one image of each person, all taken under the same conditions of lighting, image size, and head orientation. All images in the database were then classified as being one of these sixteen individuals (i.e., the threshold θ_ϵ was effectively infinite, so that no faces were rejected as unknown). Seven eigenfaces were used in the classification process.

Statistics were collected measuring the mean accuracy as a function of the difference between the training conditions and the test conditions. The independent variables were difference in illumination, imaged head size, head orientation, and combinations of illumination, size, and orientation.

Figure 9 shows results of these experiments for the case of infinite θ_ϵ. The graphs of the figure show the number of correct classifications for varying conditions of lighting, size, and head orientation, averaged over the number of experiments. For this case where every face image is classified as known, the system achieved approximately 96% correct classification averaged over lighting variation, 85% correct averaged over orientation variation, and 64% correct averaged over size variation.

As can be seen from these graphs, changing lighting conditions causes relatively few errors, while performance drops dramatically with size change. This is not surprising, since under lighting changes alone the neighborhood pixel correlation remains high, but under size changes the correlation from one image to another is largely lost. It is clear that there is a need for a multiscale approach, so that faces at a particular size are compared with one another. One method of accomplishing this is to make sure that each "face class" includes images of the individual at several different sizes, as was discussed in the section on Other Issues.

In a second experiment the same procedures were followed, but the acceptance threshold θ_ϵ was also varied. At low values of θ_ϵ, only images that project very closely to the known face classes will be recognized, so that there will be few errors but many of the images will be rejected as unknown. At high values of θ_ϵ most images will be classified, but there will be more errors. Adjusting θ_ϵ to achieve 100% accurate recognition boosted the unknown rates to 19% while varying lighting, 39% for orientation, and 60% for size. Setting the unknown rate arbitrarily to 20% resulted in correct recognition rates of 100%, 94%, and 74% respectively.

These experiments show an increase of performance accuracy as the threshold decreases. This can be tuned to achieve effectively perfect recognition as the threshold tends to zero, but at the cost of many images being rejected as unknown. The tradeoff between rejection rate and recognition accuracy will be different for each of the various face recognition applications. However, what would be most desirable is to have a way of setting the threshold high, so that few known face images are rejected as unknown, while at the same time detecting the incorrect classifications. That is, we would like to increase the efficiency (the d-prime) of the recognition process.

One way of accomplishing this is to also examine the (normalized) Euclidian distance between an image and face space as a whole. Because the projection onto the eigenface vectors is a many-to-one mapping, there is a potentially unlimited number of images that can project onto the eigenfaces in the same manner, i.e., produce the same weights. Many of these will look nothing like a face, as shown in Figure 4c. This approach was described in the section on Using "Face Space" to Locate the Face as a method of identifying likely face subimages.

Real-Time Recognition

We have used the techniques described above to build a system that locates and recognizes faces in near-real-time in a reasonably unstructured environment. Figure 10 shows a diagram of the system. A fixed camera, monitoring part of a room, is connected to a Datacube image

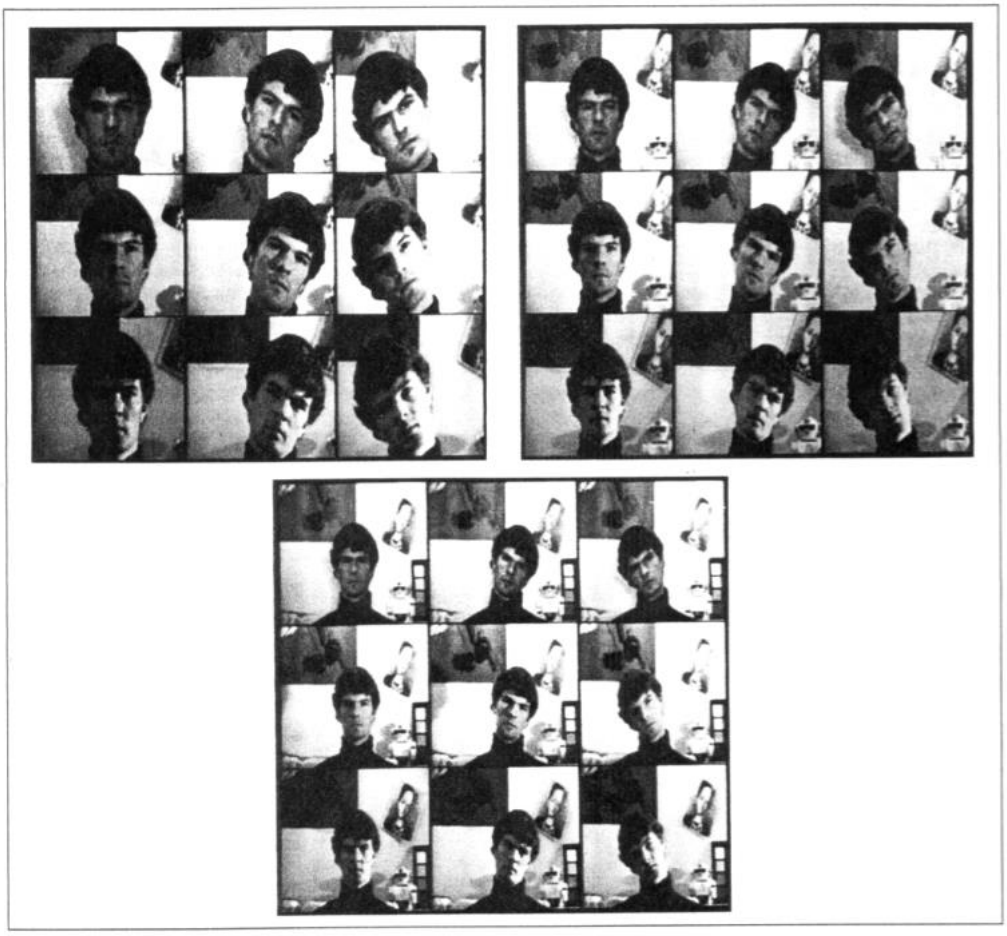

Figure 8. Variation of face images for one individual: three head sizes, three lighting conditions, and three head orientations.

processing system, which resides on the bus of a Sun 3/160. The Datacube digitizes the video image and performs spatiotemporal filtering, thresholding, and subsampling at frame rate (30 frames/sec). (The images are subsampled to speed up the motion analysis.)

The motion detection and analysis programs run on the Sun 3/160, first detecting a moving object and then tracking the motion and applying simple rules to determine if it is tracking a head. When a head is found, the subimage, centered on the head, is sent to another computer (a Sun Sparcstation) that is running the face recognition program (although it could be running on the same computer as the motion program). Using the distance-from-face-space measure, the image is either rejected as not a face, recognized as one of a group of familiar faces, or determined to be an unknown face.

Recognition occurs in this system at rates of up to two or three times per second. Until motion is detected, or as long as the image is not perceived to be a face, there is no output. When a face is recognized, the image of the identified individual is displayed on the Sun monitor.

RELATIONSHIP TO BIOLOGY AND NEURAL NETWORKS

Biological Motivations

High-level recognition tasks are typically modeled as requiring many stages of processing, e.g., the Marr (1982)

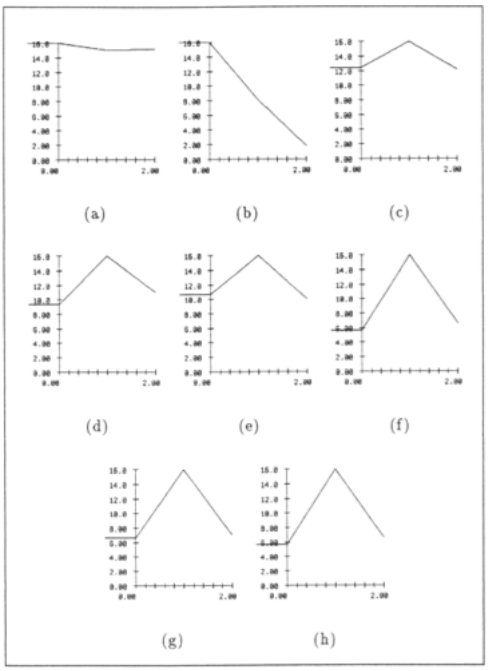

Figure 9. Results of experiments measuring recognition performance using eigenfaces. Each graph shows averaged performance as the lighting conditions, head size, and head orientation vary—the *y*-axis depicts number of correct classifications (out of 16). The peak (16/16 correct) in each graph results from recognizing the particular training set perfectly. The other two graph points reveal the decline in performance as the following parameters are varied: **(a)** lighting, **(b)** head size (scale), **(c)** orientation, **(d)** orientation and lighting, **(e)** orientation and size (#1), **(f)** orientation and size (#2), **(g)** size and lighting, **(h)** size and lighting (#2).

paradigm of progressing from images to surfaces to three-dimensional models to matched models. However, the early development and the extreme rapidity of face recognition makes it appear likely that there must also be a recognition mechanism based on some fast, low-level, two-dimensional image processing.

On strictly phenomenological grounds, such a face recognition mechanism is plausible because faces are typically seen in a limited range of views, and are a very important stimulus for humans from birth. The existence of such a mechanism is also supported by the results of a number of physiological experiments in monkey cortex claiming to isolate neurons that respond selectively to faces (e.g., see Perrett, Rolls, & Caan, 1982; Perrett, Mistlin, & Chitty, 1987; Bruce, Desimone, & Gross, 1981; Desimone, Albright, Gross, & Bruce, 1984; Rolls, Baylis, Hasselmo, & Nalwa, 1989). In these experiments, some cells were sensitive to identity, some to "faceness," and some only to particular views (such as frontal or profile).

Although we do not claim that biological systems have "eigenface cells" or process faces in the same way as the eigenface approach, there are a number of qualitative similarities between our approach and current understanding of human face recognition. For instance, relatively small changes cause the recognition to degrade gracefully, so that partially occluded faces can be recognized, as has been demonstrated in single-cell recording experiments. Gradual changes due to aging are easily handled by the occasional recalculation of the eigenfaces, so that the system is quite tolerant to even large changes as long as they occur over a long period of time. If, however, a large change occurs quickly—e.g., addition of a disguise or change of facial hair—then the eigenfaces approach will be fooled, as are people in conditions of casual observation.

Neural Networks

Although we have presented the eigenfaces approach to face recognition as an information-processing model, it may be implemented using simple parallel computing elements, as in a connectionist system or artificial neural network. Figure 11 shows a three-layer, fully connected linear network that implements a significant part of the system. The input layer receives the input (centered and normalized) face image, with one element per image pixel, or N elements. The weights from the input layer to the hidden layer correspond to the eigenfaces, so that the value of each hidden unit is the dot product of the input image and the corresponding eigenface: $\omega_l = \mathbf{\Phi}^T \mathbf{u}_l$. The hidden units, then, form the pattern vector $\mathbf{\Omega}^T = [\omega_1, \omega_2 \ldots \omega_L]$.

The output layer produces the face space projection of the input image when the output weights also correspond to the eigenfaces (mirroring the input weights). Adding two nonlinear components we construct Figure 12, which produces the pattern class $\mathbf{\Omega}$, face space projection $\mathbf{\Phi}_f$, distance measure d (between the image and its face space projection), and a classification vector. The classification vector is comprised of a unit for each known face defining the pattern space distances ϵ_l. The unit with the smallest value, if below the specified threshold θ_ϵ, reveals the identity of the input face image.

Parts of the network of Figure 12 are similar to the associative networks of Kohonen (1989) and Kohonen and Lehtio (1981). These networks implement a learned stimulus–response mapping, in which the learning phase modifies the connection weights. An autoassociative network implements the projection onto face space. Similarly, reconstruction using eigenfaces can be used to recall a partially occluded face, as shown in Figure 13.

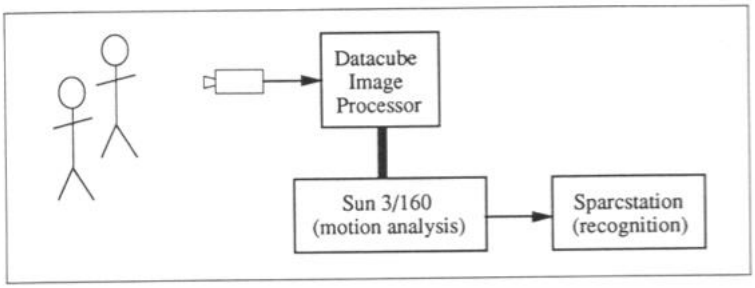

Figure 10. System diagram of the face recognition system.

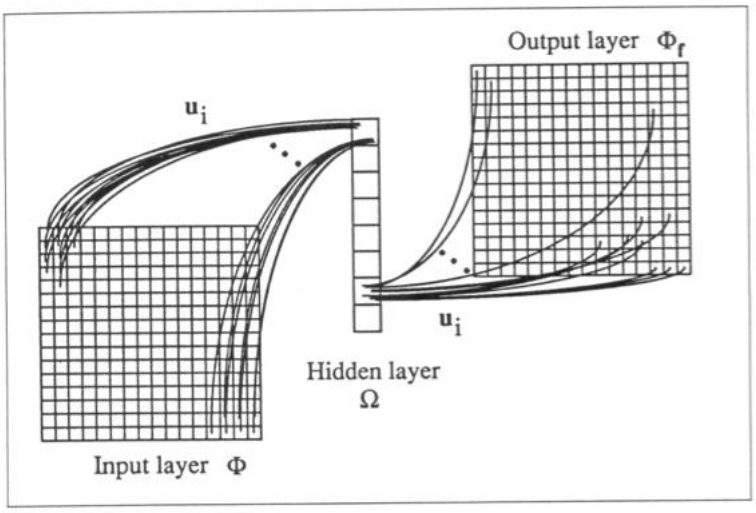

Figure 11. Three-layer linear network for eigenface calculation. The symmetric weights $\mathbf{u}_i$ are the eigenfaces, and the hidden units reveal the projection of the input image Φ onto the eigenfaces. The output Φ_f is the face space projection of the input image.

CONCLUSION

Early attempts at making computers recognize faces were limited by the use of impoverished face models and feature descriptions (e.g., locating features from an edge image and matching simple distances and ratios), assuming that a face is no more than the sum of its parts, the individual features. Recent attempts using parameterized feature models and multiscale matching look more promising, but still face severe problems before they are generally applicable. Current connectionist approaches tend to hide much of the pertinent information in the weights that makes it difficult to modify and evaluate parts of the approach.

The eigenface approach to face recognition was motivated by information theory, leading to the idea of basing face recognition on a small set of image features that best approximates the set of known face images, without requiring that they correspond to our intuitive notions of facial parts and features. Although it is not an elegant solution to the general recognition problem, the eigenface approach does provide a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple, and has been shown to work well in a constrained environment. It can also be implemented using modules of connectionist or neural networks.

It is important to note that many applications of face recognition do not require perfect identification, although most require a low false-positive rate. In searching a large database of faces, for example, it may be preferable to find a small set of likely matches to present to the user. For applications such as security systems or human–computer interaction, the system will normally be able to "view" the subject for a few seconds or minutes, and thus will have a number of chances to recognize the person. Our experiments show that the eigenface technique can be made to perform at very high accuracy, although with a substantial "unknown" rejection rate, and thus is potentially well suited to these applications.

We are currently investigating in more detail the issues of robustness to changes in lighting, head size, and head orientation, automatically learning new faces, incorpo-

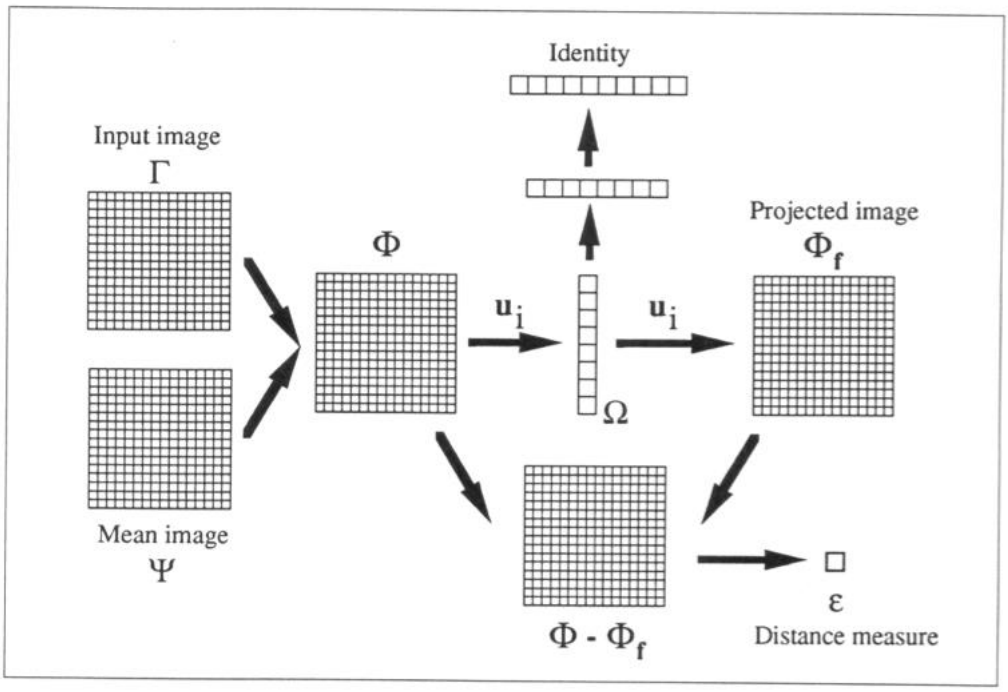

Figure 12. Collection of networks to implement computation of the pattern vector, projection into face space, distance from face space measure, and identification.

Figure 13. **(a)** Partially occluded face image and **(b)** its reconstruction using the eigenfaces.

rating a limited number of characteristic views for each individual, and the tradeoffs between the number of people the system needs to recognize and the number of eigenfaces necessary for unambiguous classification. In addition to recognizing faces, we are also beginning efforts to use eigenface analysis to determine the gender of the subject and to interpret facial expressions, two important face processing problems that complement the task of face recognition.

REFERENCES

Bledsoe, W. W. (1966a). The model method in facial recognition. Panoramic Research Inc., Palo Alto, CA, Rep. PRI:15, August.

Bledsoe, W. W. (1966b). Man-machine facial recognition. Panoramic Research Inc., Palo Alto, CA, Rep. PRI:22, August.

Bruce, V. (1988). *Recognising faces.* Hillsdale, NJ: Erlbaum.

Bruce, C. J., Desimone, R., & Gross, C. G. (1981). *Journal of Neurophysiology, 46,* 369–384.

Burt, P. (1988a). Algorithms and architectures for smart sensing. *Proceedings of the Image Understanding Workshop,* April.

Burt, P. (1988b). Smart sensing within a Pyramid Vision Machine. *Proceedings of IEEE, 76*(8), 139–153.

Cannon, S. R., Jones, G. W., Campbell, R., & Morgan, N. W. (1986). A computer vision system for identification of individuals. *Proceedings of IECON, 1.*

Carey, S., & Diamond, R. (1977). From piecemeal to configurational representation of faces. *Science, 195,* 312–313.

Craw, Ellis, & Lishman (1987). Automatic extraction of face features. *Pattern Recognition Letters, 5,* 183–187.

Davies, Ellis, & Shepherd (Eds.), (1981). *Perceiving and remembering faces.* London: Academic Press.

Desimone, R., Albright, T. D., Gross, C. G., & Bruce, C. J. (1984). Stimulus-selective properties of inferior temporal neurons in the macaque. *Neuroscience, 4,* 2051–2068.

Fischler, M. A., & Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers, c-22*(1).

Fleming, M., & Cottrell, G. (1990). Categorization of faces using unsupervised feature extraction. *Proceedings of IJCNN-90, 2.*

Goldstein, Harmon, & Lesk (1971). Identification of human faces. *Proceedings IEEE, 59,* 748.

Harmon, L. D. (1971). Some aspects of recognition of human faces. In O. J. Grusser & R. Klinke (Eds.), *Pattern recognition in biological and technical systems.* Berlin: Springer-Verlag.

Hay, D. C., & Young, A. W. (1982). The human face. In A. W. Ellis (Ed.), *Normality and pathology in cognitive functions.* London: Academic Press.

Kanade, T. (1973). Picture processing system by computer complex and recognition of human faces. Dept. of Information Science, Kyoto University.

Kaya, Y., & Kobayashi, K. (1972). A basic study on human face recognition. In S. Watanabe (Ed.), *Frontiers of pattern recognition.* New York: Academic Press.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen–Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(1).

Kohonen, T. (1989). *Self-organization and associative memory.* Berlin: Springer-Verlag.

Kohonen, T., & Lehtio, P. (1981). Storage and processing of information in distributed associative memory systems. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum, pp. 105–143.

Marr, D. (1982). *Vision.* San Francisco: W. H. Freeman.

Midorikawa, H. (1988). The face pattern identification by back-propagation learning procedure. *Abstracts of the First Annual INNS Meeting,* Boston, p. 515.

O'Toole, Millward, & Anderson (1988). A physical system approach to recognition memory for spatially transformed faces. *Neural Networks, 1,* 179–199.

Perrett, D. I., Mistlin, A. J., & Chitty, A. J. (1987). Visual neurones responsive to faces. *TINS, 10*(9), 358–364.

Perrett, Rolls, & Caan (1982). Visual neurones responsive to faces in the monkey temporal cortex. *Experimental Brain Research, 47,* 329–342.

Rolls, E. T., Baylis, G. C., Hasselmo, M. E., & Nalwa, V. (1989). The effect of learning on the face selective responses of neurons in the cortex in the superior temporal sulcus of the monkey. *Experimental Brain Research, 76,* 153–164.

Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A, 4*(3), 519–524.

Stonham, T. J. (1986). Practical face recognition and verification with WISARD. In H. Ellis, M. Jeeves, F. Newcombe, & A. Young (Eds.), *Aspects of face processing.* Dordrecht: Martinus Nijhoff.

Wong, K., Law, H., & Tsang, P. (1989). A system for recognising human faces. *Proceedings of ICASSP,* May, 1638–1642.

Yuille, A. L., Cohen, D. S., & Hallinan, P. W. (1989). Feature extraction from faces using deformable templates. *Proceedings of CVPR,* San Diego, CA, June.