Face Recognition Using Eigenfaces

Matthew A. Turk and Alex P. Pentland

Vision and Modeling Group, The Media Laboratory
Massachusetts Institute of Technology

Abstract

We present an approach to the detection and identification of human faces and describe a working, near-real-time face recognition system which tracks a subject's head and then recognizes the person by comparing characteristics of the face to those of known individuals. Our approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are are normally upright and thus may be described by a small set of 2-D characteristic views. Face images are projected onto a feature space ("face space") that best encodes the variation among known face images. The face space is defined by the "eigenfaces", which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.

1 Introduction

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[1]. 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 towards developing a sort of early, preattentive pattern recognition capability that does not depend upon having full three-dimensional models or detailed geometry. Our aim was to develop a computational model of face recognition which is fast, reasonably simple, and accurate in constrained environments such as an office or a household.

Although face recognition is a high level visual problem, there is quite a bit of structure imposed on the task. We take advantage of some of this structure by proposing a scheme for recognition which is based on an information theory approach, seeking to encode the most relevant information in a group of faces which will best distinguish them from one another. The approach transforms face images into a small set of characteristic feature images, called "eigenfaces", which are 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 reasonably 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 relative insensitivity to small or gradual changes in the face image.

1.1 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. Beginning with Bledsoe's [2] and Kanade's [3] early systems, a number of automated or semi-automated face recognition strategies have modeled and classified faces based on normalized distances and ratios among feature points. Recently this general approach has been continued and improved by the recent work of Yuille *et al.* [4].

Such approaches have proven difficult to extend to multiple views, and have often been quite fragile. 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 [5]. Nonetheless, this approach to face recognition remains the most popular one in the computer vision literature.

Connectionist approaches to face identification seek to capture the configurational, or gestalt-like nature of the task. Fleming and Cottrell [6], building on earlier work by Kohonen and Lahtio [7], use nonlinear units to train a network via back propagation to classify face images. Stonham's WISARD system [8] has been applied with some success to binary face images, recognizing both identity and expression. Most connectionist systems dealing with faces treat the input image as a general 2-D pattern,

and can make no explicit use of the configurational properties of a face. Only very simple systems have been explored to date, and it is unclear how they will scale to larger problems.

Recent work by Burt *et al.* uses a "smart sensing" approach based on multiresolution template matching [9]. 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. The face models are built by hand from face images.

2 Eigenfaces for Recognition

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, assuming that predefined measurements were relevant and sufficient. 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.

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 judgement 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. These eigenvectors can be thought of as a set of features which 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 these faces are shown in Figure 2.

Each face image in the training set can be represented exactly in terms of a linear combination of the eigenfaces. The number of possible eigenfaces is equal to the number of face images in the training set. However the faces 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 primary reason for using fewer eigenfaces is computational efficiency. The best M' eigenfaces span an M'-dimensional subspace — "face space" — of all possible images. As sinusoids of varying frequency and phase are the basis functions of a fourier decomposition (and are in fact eigenfunctions of linear systems), the eigenfaces are the basis vectors of the eigenface decomposition.

The idea of using eigenfaces was motivated by a technique developed by Sirovich and Kirby [10] for efficiently representing pictures of faces using principal component analysis. They argued that a 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.

It occurred to us that if a multitude of face images can be reconstructed by weighted sums of a small collection of characteristic images, then an efficient way to learn and recognize faces might be to build the characteristic features from known face images and to recognize particular faces by comparing the feature weights needed to (approximately) reconstruct them with the weights associated with the known individuals.

The following steps summarize the recognition process:

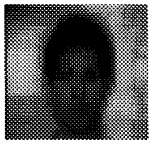
1. Initialization: Acquire the training set of face images and calculate the eigenfaces, which define the *face space*.
2. When a new face image is encountered, calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.
3. Determine 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."
4. If it is a face, classify the weight pattern as either a known person or as unknown.
5. (Optional) If the same unknown face is seen several times, calculate its characteristic weight pattern and incorporate into the known faces (i.e., learn to recognize it).

2.1 Calculating Eigenfaces

Let a face image $I(x, y)$ be a two-dimensional N by N array of intensity values, or a vector of dimension N^2. A typical image of size 256 by 256 describes 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 component analysis (or Karhunen-Loeve expansion) is to find the vectors which 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.

(a)



(b)

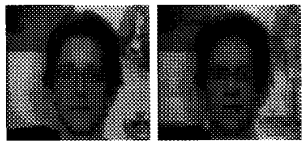
Figure 1: (a) Face images used as the training set. (b) The average face Ψ.

Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, ... \Gamma_M$. 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 1(a), with the average face Ψ shown in Figure 1(b). This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors $\mathbf{u}_n$ and their associated eigenvalues λ_k which best describes the distribution of the data. The vectors $\mathbf{u}_k$ and scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix

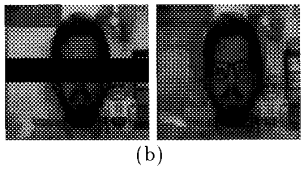
$$\begin{aligned} C &= \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T \\ &= AA^T \end{aligned} \qquad (1)$$

where the matrix $A = [\,\Phi_1\ \Phi_2\ ...\ \Phi_M\,]$. 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. Fortunately we can determine the eigenvectors by first solving a much smaller M by M matrix problem, and taking linear combinations of the resulting vectors. (See [11] for the details.)

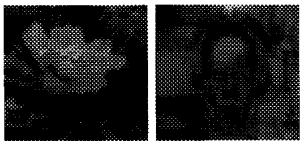
With this analysis the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the training set of face images will be relatively small ($M \ll N^2$), and the calculations become quite manageable. 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. Normally the background is removed by cropping training images, so that the eigenfaces have zero values outside of the face area.



(a)



(b)



(c)

Figure 3: Three images and their projections onto the face space defined by the eigenfaces of Figure 2.

2.2 Using Eigenfaces to classify a face image

Once the eigenfaces are created, 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. The number of eigenfaces to be used is chosen heuristically based on the eigenvalues.

A new face image (Γ) is transformed into its eigenface components (projected into "face space") by a simple operation, $\omega_k = \mathbf{u}_k^T(\Gamma - \Psi)$ for $k = 1, \ldots, M'$. This describes a set of point-by-point image multiplications and summations. Figure 3 shows three images and their projections into the seven-dimensional face space.

The weights form a vector $\Omega^T = [\omega_1\ \omega_2\ \ldots\ \omega_{M'}]$ that describes the contribution of each eigenface in representing the input face image, treating the

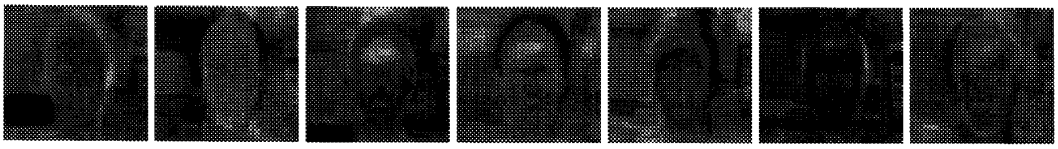


Figure 2: Seven of the eigenfaces calculated from the images of Figure 1, without the background removed.

eigenfaces as a basis set for face images. The vector is used to find which of a number of pre-defined 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 $\epsilon_k = \|(\Omega - \Omega_k)\|$, where Ω_k is a vector describing the kth face class. 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".

2.3 Using Eigenfaces to detect faces

We can also use knowledge of the face space to detect and locate faces in single images. This allows us to recognize the presence of faces apart from the task of identifying them.

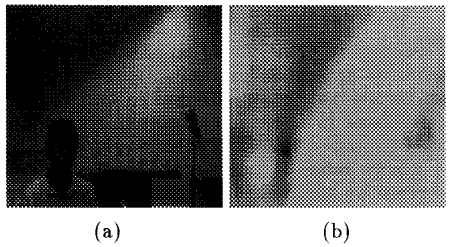
Creating the vector of weights for an image is equivalent to projecting the image onto the low-dimensional face space. The distance ε between the image and its projection onto the face space is simply the distance between the mean-adjusted input image $\Phi = \Gamma - \Psi$ and $\Phi_f = \sum_{i=1}^{i=M'} \omega_k \mathbf{u}_k$, its projection onto face space.

As seen in Figure 3, images of faces do not change radically when projected into the face space, while the projection of non-face images appear 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" $\varepsilon(x, y)$. Figure 4 shows an image and its face map — low values (the dark area) indicate the presence of a face. There is a distinct minimum in the face map corresponding to the location of the face in the image.

Unfortunately, direct calculation of this distance measure is rather expensive. We have therefore developed a simpler, more efficient method of calculating the face map $\varepsilon(x, y)$, which is described in [11].

2.4 Face space revisited

An image of a face, and in particular the faces in the training set, should lie near the face space, which in general describes images that are "face-like". In other words, the projection distance ε should be within some threshold θ_δ. Images of known individuals should project to near the corresponding face class, i.e. $\epsilon_k < \theta_\epsilon$. 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. Figure 5 shows these four options for the simple example of two eigenfaces.



(a) (b)

Figure 4: (a) Original image. (b) Face map, where low values (dark areas) indicate the presence of a face.

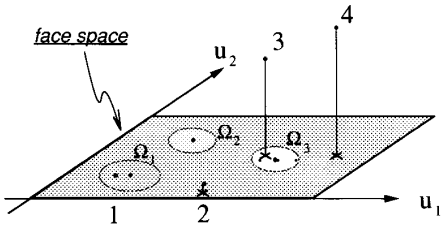


Figure 5: A simplified version of face space to illustrate the four results of projecting an image into face space. In this case, there are two eigenfaces ($\mathbf{u}_1$ and $\mathbf{u}_2$) and three known individuals (Ω_1, Ω_2, and Ω_3).

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 3 shows some images and their projections into face space. Figure 3 (a) and (b) are examples of case 1, while Figure 3 (c) illustrates case 4.

In our current system calculation of the eigenfaces is done offline as part of the training. The recognition currently takes about 350 msec running rather inefficiently in Lisp on a Sun Sparcstation 1, using face images of size 128x128.

3 Recognition Experiments

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 images from Figure 1(a) 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 512x512 pixels down to 16x16 pixels.

In the first experiment the effects of varying lighting, size, and head orientation were investigated using the complete database of 2500 images. Various groups of sixteen 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 — no faces were rejected as unknown.

Statistics were collected measuring the mean accuracy as a function of the difference between the training conditions and the test conditions. In the case of infinite θ_ϵ and θ_δ, the system achieved approximately 96% correct classification averaged over lighting variation, 85% correct averaged over orientation variation, and 64% correct averaged over size variation.

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 which project very closely to the known face classes (cases 1 and 3 in Figure 5) 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.

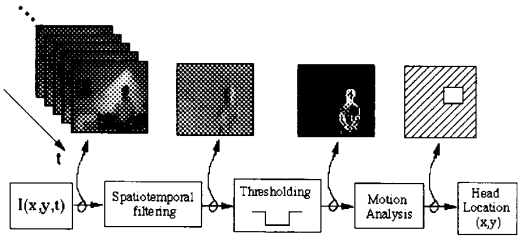


Figure 6: The head tracking and locating system.

These experiments show an increase of performance accuracy as the acceptance 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.

The results also indicate that 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 quite low. It is clear that there is a need for a multiscale approach, so that faces at a particular size are compared with one another.

4 Real-time recognition

People are constantly moving. Even while sitting, we fidget and adjust our body position, blink, look around, and such. For the case of a moving person in a static environment, we built a simple motion detection and tracking system, depicted in Figure 6, which locates and tracks the position of the head. Simple spatio-temporal filtering followed by a nonlinearity accentuates image locations that change in intensity over time, so a moving person "lights up" in the filtered image.

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 aren't expected to jump around the image erratically). Figure 7 shows an image with the head located, along with the path of the head in the preceding sequence of frames.

We have used the techniques described above to build a system which locates and recognizes faces in near-real-time in a reasonably unstructured environment. When the motion detection and analysis

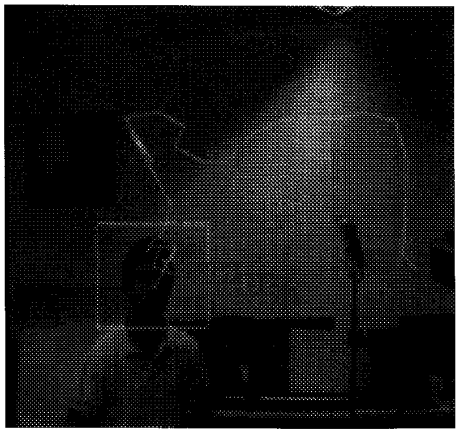


Figure 7: The head has been located — the image in the box is sent to the face recognition process. Also shown is the path of the head tracked over several previous frames.

programs finds a head, a subimage, centered on the head, is sent to the face recognition module. Using the distance-from-face-space measure ϵ, 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.

5 Further Issues and Conclusion

We are currently extending the system to deal with a range of aspects (other than full frontal views) by defining a small number of face classes for each known person corresponding to characteristic views. Because of the speed of the recognition, the system has many chances within a few seconds to attempt to recognize many slightly different views, at least one of which is likely to fall close to one of the characteristic views.

An intelligent system should also have an ability to adapt over time. Reasoning about images in face space provides a means to learn and subsequently recognize new faces in an unsupervised manner. When an image is sufficiently close to face space (i.e., it is face-like) 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.

A noisy image or partially occluded face should cause recognition performance to degrade gracefully., since the system essentially implements an autoassociative memory for the known faces (as described in [7]). This is evidenced by the projection of the occluded face image of Figure 3(b).

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 approximate 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 object 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 somewhat constrained environment.

References

[1] Davies, Ellis, and Shepherd (eds.), *Perceiving and Remembering Faces*, Academic Press, London, 1981.

[2] W. W. Bledsoe, "The model method in facial recognition," Panoramic Research Inc., Palo Alto, CA, Rep. PRI:15, Aug. 1966.

[3] T. Kanade, "Picture processing system by computer complex and recognition of human faces," Dept. of Information Science, Kyoto University, Nov. 1973.

[4] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," *Proc. CVPR*, San Diego, CA, June 1989.

[5] S. Carey and R. Diamond, "From piecemeal to configurational representation of faces," *Science*, Vol. 195, Jan. 21, 1977, 312-13.

[6] M. Fleming and G. Cottrell, "Categorization of faces using unsupervised feature extraction," *Proc. IJCNN-90*, Vol. 2.

[7] T. Kohonen and P. Lehtio, "Storage and processing of information in distributed associative memory systems," in G. E. Hinton and J. A. Anderson, *Parallel Models of Associative Memory*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1981, pp. 105-143.

[8] T. J. Stonham, "Practical face recognition and verification with WISARD," in H. Ellis, M. Jeeves, F. Newcombe, and A. Young (eds.), *Aspects of Face Processing*, Martinus Nijhoff Publishers, Dordrecht, 1986.

[9] P. Burt, "Smart sensing within a Pyramid Vision Machine," *Proc. IEEE*, Vol. 76, No. 8, Aug. 1988.

[10] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Am. A*, Vol. 4, No. 3, March 1987, 519-524.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, March 1991.