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 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") and are compared to the face space of 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
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 contains 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. These eigenvectors can be thought of as a set of features which together characterize the

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 $N^2$. The first step is to center the face images, i.e., subtract the mean intensity value from each pixel so the result is a set of $N^2$ vectors of zero mean:

$$
I_c(x,y) = I(x,y) - \overline{I},
$$

where $\overline{I}$ is the average intensity of all pixels in the image.

The next step is to calculate the $N^2$ by $N^2$ covariance matrix $C$ of the centered images. This is a symmetric matrix where the $(i,j)$-th entry is

$$
C_{ij} = \frac{1}{M} \sum_{k=1}^{M} (I_c(x_k,y_k) - \overline{I_c})(I_c(x_j,y_j) - \overline{I_c})
$$

and the diagonal entries are all zero. The eigenvalues $\lambda_i$ and eigenvectors $\varphi_i$ of $C$ are calculated.

The eigenvector $\varphi_i$ corresponding to the largest eigenvalue $\lambda_i$ is the 1st eigenface, the 2nd largest eigenvalue $\lambda_2$ corresponds to the 2nd eigenface, and so on. The eigenfaces are standardized by dividing by the square root of their eigenvalues.

The face images are then projected onto each of the eigenfaces to get a set of weights $w_i$ for each face:

$$
I(x,y) = \sum_{i=1}^{M} w_i \varphi_i(x,y)
$$

where $M$ is the number of eigenfaces used for reconstruction. The weights $w_i$ are stored for later use in recognition.

The recognition process is as follows:

1. For a given input face image $I(x,y)$, calculate the weights $w_i$ as above.
2. Find the face in the database with the smallest distance (using the Euclidean distance) between the calculated weights $w_i$ and the weights $w_{i_0}$ of the face in the database.
3. The face with the smallest distance is the recognized face.

This method is efficient because it reduces the dimensionality of the problem and allows for fast recognition of faces.
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 2000 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 belonging to 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 $\theta_a$ and $\theta_z$, 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 $\theta_a$ was also varied. At low values of $\theta_a$, 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 $\theta_a$, most images will be classified, but there will be more errors. Adjusting $\theta_a$ 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 effective 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 non-linearity 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 z, 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.

Further Issues and Conclusions

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


