Obtaining the Eigenface Basis*

Jon Krueger
Marshall B. Robinson
Doug Kochelek
Matthew Escarra

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

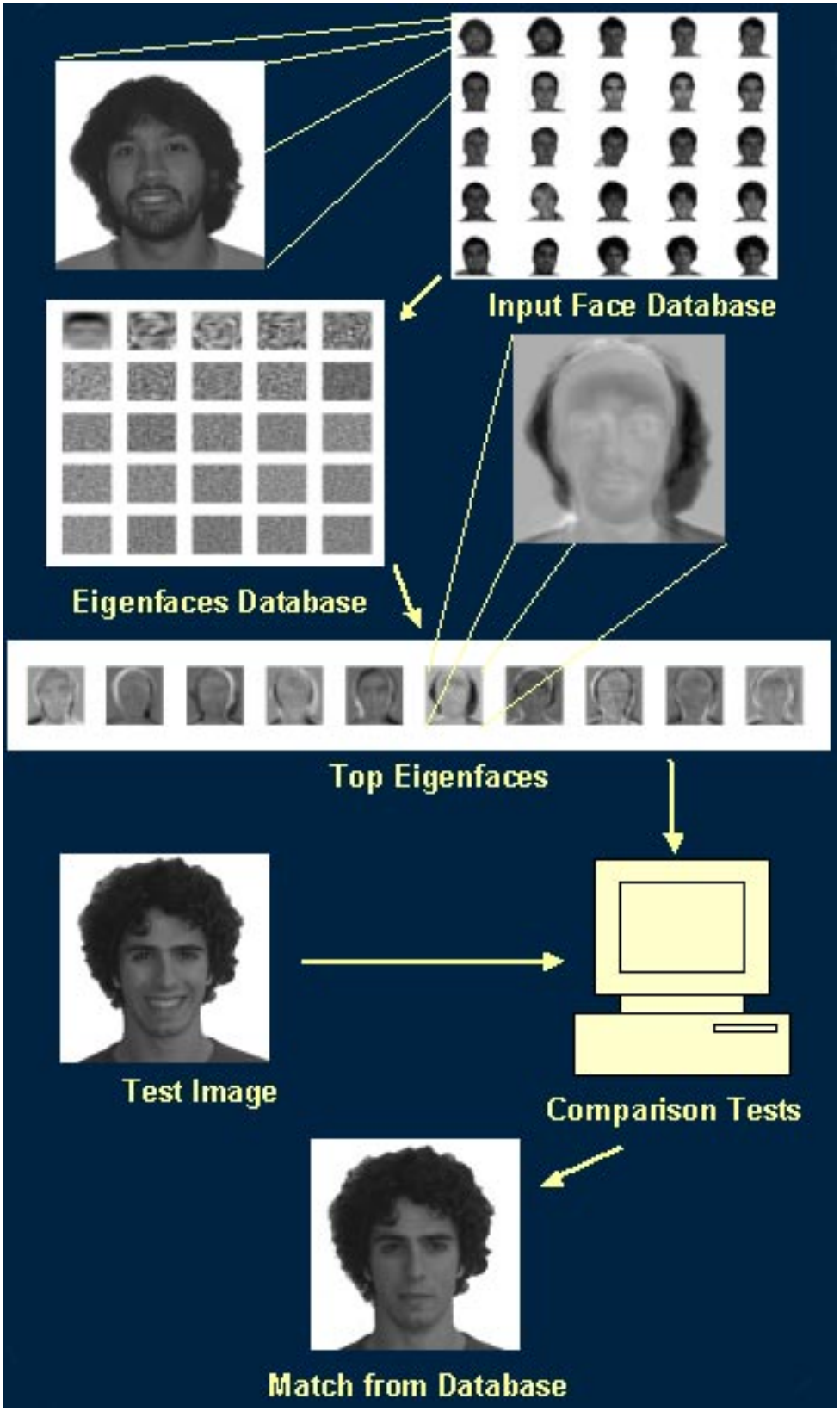
This page explains how to set up our face recognition system for detection. It is centered around the creation of the "eigenface" basis for "face space." It also discusses simplifying the eigenface basis to a level that is both managable and accurate.

1 Introduction to Eigenface System

The eigenface face recognition system can be divided into two main segments: creation of the eigenface basis and recognition, or detection, of a new face. The system follows the following general flow:

*Version 1.3: Dec 17, 2004 11:04 pm -0600
†http://creativecommons.org/licenses/by/1.0

Summary of Overall Face Recognition Process



http://cnx.org/content/m12531/1.3/

Figure 1: A robust detection system can yield correct matches when the person is feeling happy or sad.

2 Deriving the Eigenface Basis

The eigenface technique is a powerful yet simple solution to the face recognition dilemma. In fact, it is really the most intuitive way to classify a face. As we have shown, old techniques focused on particular features of the face. The eigenface technique uses much more information by classifying faces based on general facial patterns. These patterns include, but are not limited to, the specific features of the face. By using more information, eigenface analysis is naturally more effective than feature-based face recognition.

Eigenfaces are fundamentally nothing more than basis vectors for real faces. This can be related directly to one of the most fundamental concepts in electrical engineering: Fourier analysis. Fourier analysis reveals that a sum of weighted sinusoids at differing frequencies can recompose a signal perfectly! In the same way, a sum of weighted eigenfaces can seamlessly reconstruct a specific person's face.

Determining what these eigenfaces are is the crux of this technique.

Before finding the eigenfaces, we first need to collect a set of face images. These face images become our database of known faces. We will later determine whether or not an unknown face matches any of these known faces. All face images must be the same size (in pixels), and for our purposes, they must be grayscale, with values ranging from 0 to 255. Each face image is converted into a vector Γ_n of length N (N=imagewidth*imageheight). The most useful face sets have multiple images per person. This sharply increases accuracy, due to the increased information available on each known individual. We will call our collection of faces "face space." This space is of dimension N.

Example Images from the Rice Database

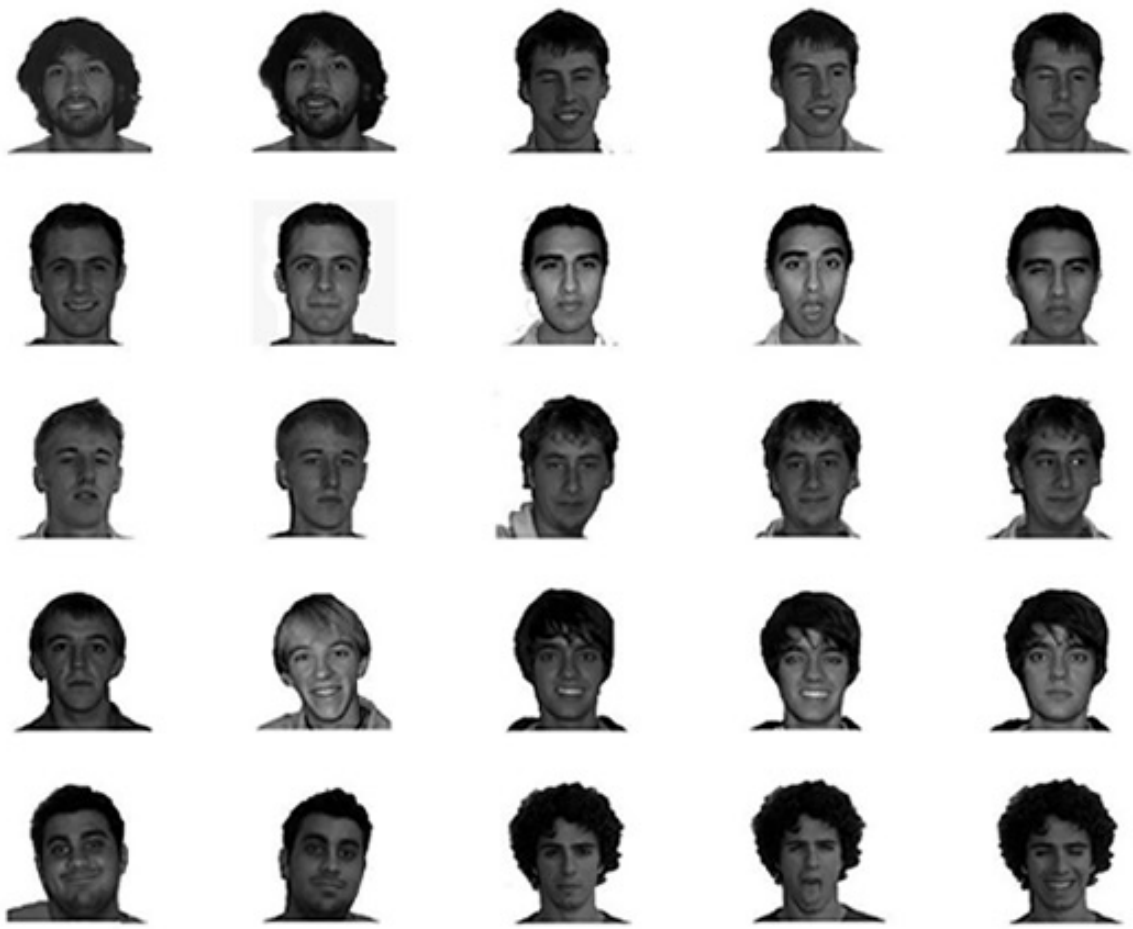


Figure 2

Next we need to calculate the average face in face space. Here M is the number of faces in our set:

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{1}$$

Average Face from Rice Database

Figure 3

We then compute each face's difference from the average:

$$\Phi_i = \Gamma_i - \Psi \tag{2}$$

We use these differences to compute a covariance matrix (C) for our dataset. The covariance between two sets of data reveals how much the sets correlate.

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = \frac{1}{M}\sum_{n=1}^{M}\begin{pmatrix} \mathrm{var}\left(p_1\right) & \dots & \mathrm{cov}\left(p_1, p_N\right) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}\left(p_N, p_1\right) & \cdots & \mathrm{var}\left(p_N\right) \end{pmatrix}_n = AA^T \tag{3}$$

Where $A = [\Phi_1\Phi_2...\Phi_M]$ and p_i = pixel i of face n.

The eigenfaces that we are looking for are simply the eigenvectors of C. However, since C is of dimension N (the number of pixels in our images), solving for the eigenfaces gets ugly very quickly. Eigenface face recognition would not be possible if we had to do this. This is where the magic behind the eigenface system happens.

3 Simplifying the Initial Eigenface Basis

Based on a statistical technique known as Principal Component Analysis (PCA), we can reduce the number of eigenvectors for our covariance matrix from N (the number of pixels in our image) to M (the number of images in our dataset). This is huge! In general, PCA is used to describe a large dimensional space with a relative small set of vectors. It is a popular technique for finding patterns in data of high dimension, and is used commonly in both face recognition and image compression.* PCA is applicable to face recognition because face images usually are very similar to each other (relative to images of non-faces) and clearly share the same general pattern and structure.

PCA tells us that since we have only M images, we have only M non-trivial eigenvectors. We can solve for these eigenvectors by taking the eigenvectors of a new M x M matrix:

$$L = A^T A \tag{4}$$

Because of the following math trick:

$$A^T A v_i = \mu_i v_i$$
$$A A^T A v_i = \mu_i A v_i$$

Where v_i is an eigenvector of L. From this simple proof we can see that Av_i is an eigenvector of C.

The M eigenvectors of L are finally used to form the M eigenvectors u_l of C that form our eigenface basis:

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \tag{5}$$

It turns out that only M-k eigenfaces are actually needed to produce a complete basis for the face space, where k is the number of unique individuals in the set of known faces.

In the end, one can get a decent reconstruction of the image using only a few eigenfaces (M'), where M' usually ranges anywhere from .1M to .2M. These correspond to the vectors with the highest eigenvalues and represent the most variance within face space.

Top Ten Eigenfaces from Rice Database

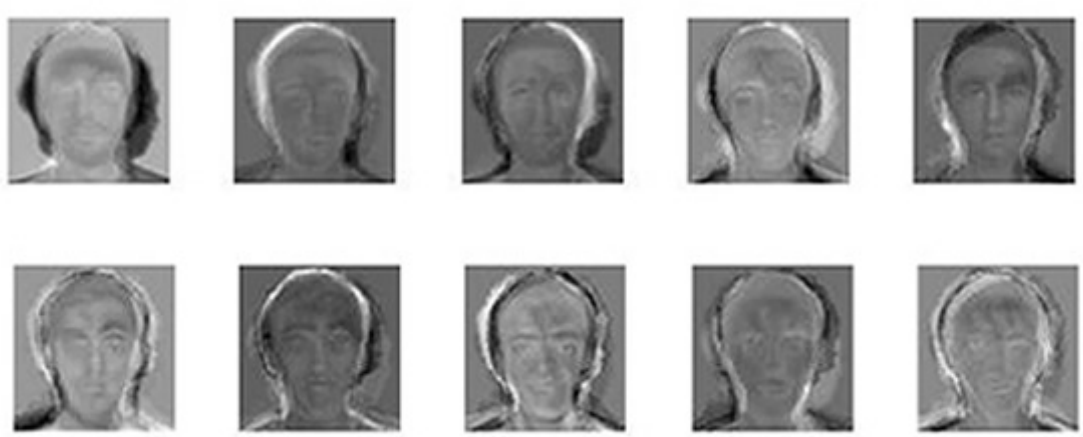


Figure 4

These eigenfaces provide a small yet powerful basis for face space. Using only a weighted sum of these eigenfaces, it is possible to reconstruct each face in the dataset. Yet the main application of eigenfaces, face recognition, takes this one step further.

*For more information on Principal Component Analysis, check out this easy to follow tutorial[1] .

[1] http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf