extensive body of research in face recognition using similar techniques. A good starting point for locating additional resources and bibliographies are available on the internet; for example, the Face Recognition Homepage [21] and MIT's Artificial Intelligence Lab [27]. A recent text [19] also gives more details on KL, its application to faces, and more applications to other data analysis problems. We also note that current research in psychology and computer science is suggesting that perhaps the actual biological processes involved in familiarity are related to KL [8].

In face recognition problems, we are given a large collection of photos of faces. In this example, we assume that all photos are the same size, that they are grayscale, that they have basically the same background, and have been centered (these and many other issues can be discussed in the classroom setting, and are at the heart of the problem in applying KL in the real world).

To be more specific, suppose our photos all have 262×294 pixels. Then each photo is a vector in $\mathbb{R}^{77,028}$. We look for a small dimensional basis, Φ, in which to represent the space of all faces in the database, so that if $\boldsymbol{x}^{(i)}$ is a face, it can be represented by its (much lower dimensional) coordinates, $\left[\boldsymbol{x}^{(i)}\right]_\Phi$.

We would like to use the KL algorithm to compute this basis, but this is a very large eigenvector problem! If we have less photos than the number of pixels, it is faster to compute:

$$X^T X = V \Lambda V^T$$

and get the vectors $\Phi = U$ from this:

$$X V \Lambda^{-1} = U$$

Either way, the columns of U are in $\mathbb{R}^{77,028}$, and we will choose only a "small" number of them. Here is the interesting translation: Since each eigenvector $\mathbf{u}_i$ is a vector in $\mathbb{R}^{77,028}$, *it can be realized as a photograph (face) since it has* 262×294 *pixels!* Thus the term "Eigenface" was coined (I'm not sure who originally came up with it), and we get a set of orthonormal "faces" by which all faces in the database can be encoded using the projection coefficients. Of course, the question remains as to how many faces one will require, and it leads us to the question of how we might compute the rank of X- as we said previously, this can be a nontrivial question- A commercial package uses 128 basis vectors [31].

Once the dimension has been set, we can numerically encode every face in terms of the weights of the linear combinations- this is the key to actual face recognition. You encode someone's face, and search the database for the closest match.

In Figure 6.1, we show a sample of 4 faces from our database of 30 faces total. In gathering the data, we were somewhat careful about overall illumination, and did some image processing to ensure that all faces had the center of the left pupil at (approximately) pixel coordinates $(97, 137)$, and the images were scaled so that there were approximately 60 pixels between pupils.[2] Initially, I

[2]This can be easily accomplished by software available on the internet, for example Paint Shop Pro at (`http://www.jasc.com/`)

Figure 6.1: A Sampling of the Original Faces. Each is 262×294 pixels of grayscale values, between 0 and 255. The eyes were set at the same position in all faces, and the backgrounds were manually masked in black.

had included faces with eyeglasses, but the primary eigenvectors picked these out as significant features, so they were eliminated from the database. Furthermore, none of the faces had significant facial hair (mustache, beard), and the backgrounds were manually masked in black (which has a grayscale value of zero). In the final database, there were 30 faces, 14 male and 16 female.

We first find the mean face, and subtract it, giving the caricatures of the face. A sample is shown in Figure 6.2, where we see the mean, and one of the mean subtracted images. The figure is a result of an automatic scaling of the mean subtracted image to grayscale values between 0 and 255 for visualization purposes.

We then compute the basis, and in Figure 6.3, we show the first several eigenfaces. There are some issues here related to how we should visualize these faces, as they are orthonormal vectors, and are not naturally related to the

Figure 6.2: The mean face is shown to the left, a caricature, or mean-subtracted face, is shown to the right.

original grayscale. We performed a linear translation so that the minimum of each vector was mapped to zero (black), and the maximum of each vector was mapped to 255 (white).

The corresponding normalized eigenvalues are shown in Figure 6.4. There is a smooth transitioning between dimensions, and no large gaps. For the purpose of this example, we take the dimension to be 10, which corresponds to a retention of 74% of the total energy.

Figure 6.5 shows several reconstructions using 10 dimensions. Even with this small number of basis vectors, the reconstructions show enough features so that they are recognizable.

We might also consider the primary planar projection of the eigenfaces. These points are shown in Figure 6.6, where the male faces are shown with asterisks, and the female faces are shown as triangles. This plot is highly suggestive- there seems to be an interesting separation of the plane based on sex (although the separation is not perfect), even though sex played absolutely no part in the algorithm- the faces were not labeled. To examine this further, consider the faces in Figure 6.7. The face on the left seems to suggest primarily male features, while the face on the right seems to suggest primarily female features. The rather curious part of this is that they are in fact the same eigenvector! That is, the face on the left is our standard translation of the second eigenvector. The face on the right is the photographic negative of the face on the right (which is the visualization of the negative eigenvector). Of course, this might be an artifact of this particular database- we'll leave further speculation to the reader.

The method of eigenfaces is at the heart of the commercial face recognition methods on the market today- for example, the package by Viisage Technology [31] is in use by the military, Boston's Logan Airport, and others. The key

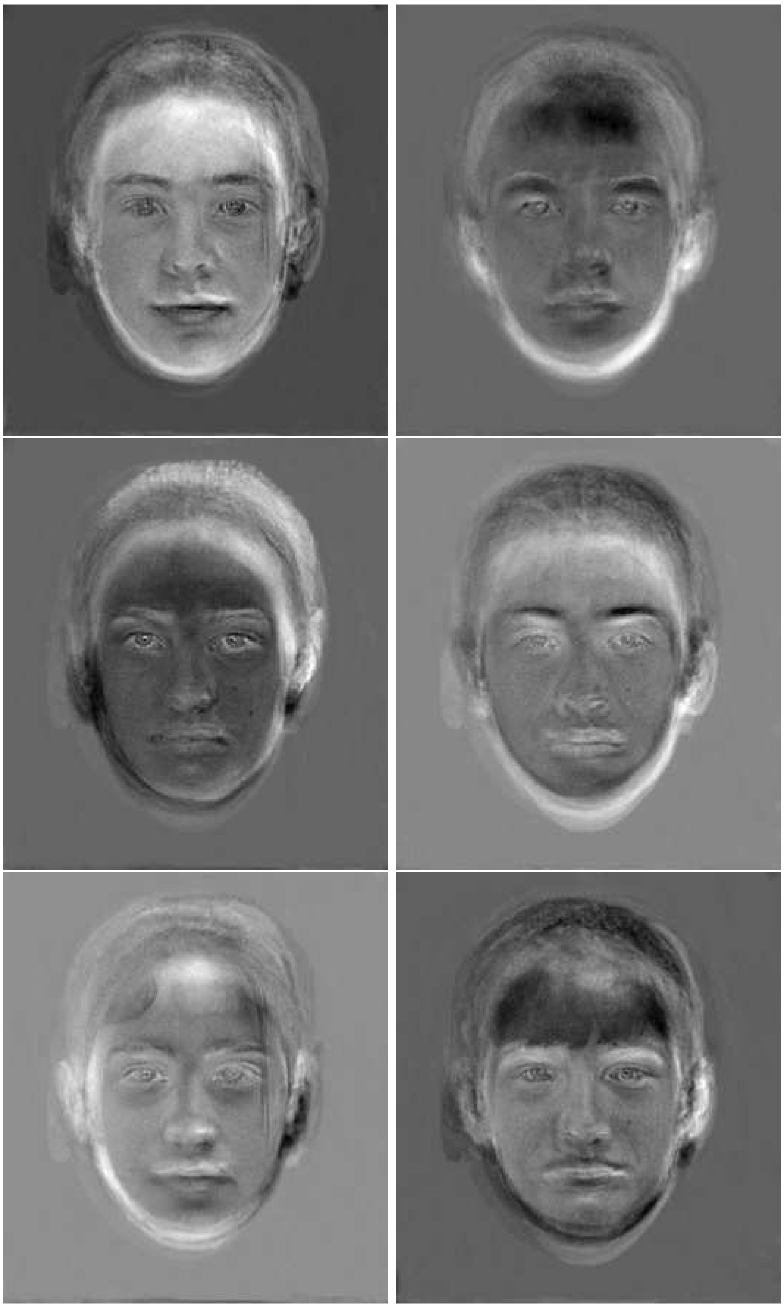

Figure 6.3: The first six Eigenfaces, which are the largest (with respect to the singular values) eigenvectors of the covariance of the data matrix.

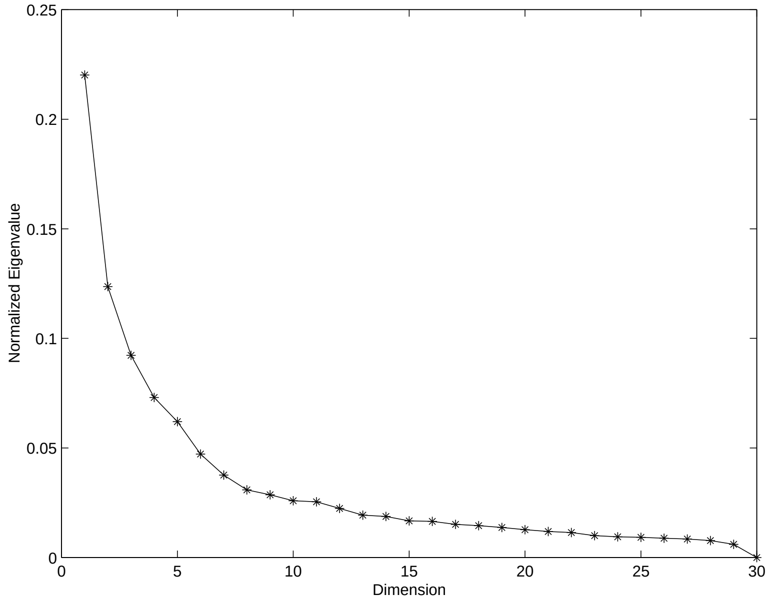

Figure 6.4: The spectrum of the covariance matrix. We see no large gaps, but fixed the dimension at 10, corresponding to retention of 74% of the total energy or variance of the system.

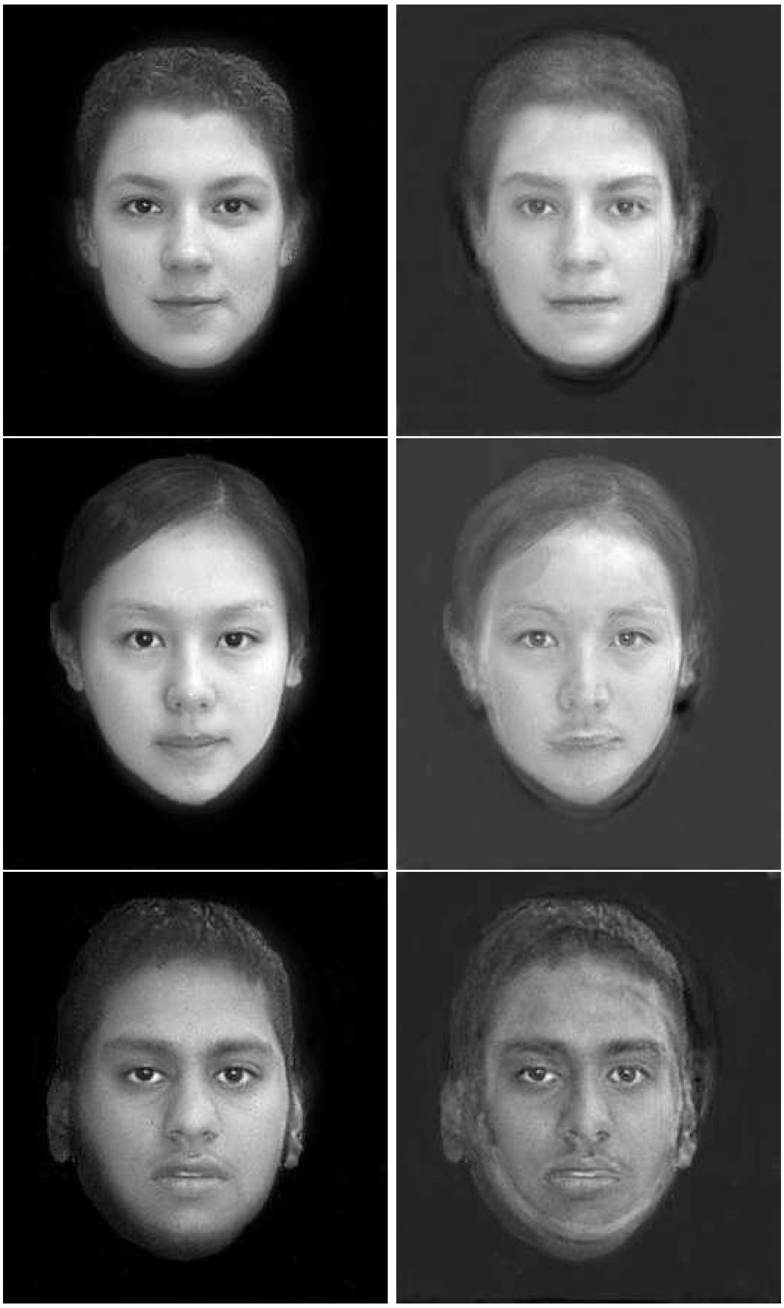

Figure 6.5: The original versus reconstructed images. The reconstructions were performed using only 10 of the basis vectors, but the reconstructions are recognizable.

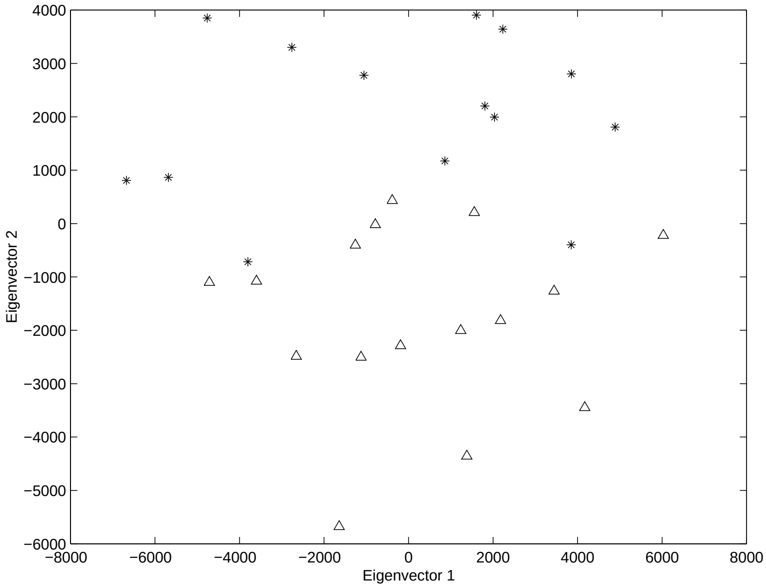

Figure 6.6: A plot of the face database to the primary plane formed by the first two eigenvectors. There seems to be an interesting separation (although not a perfect one) of the men (asterisks) and the women (triangles), and the separation is based on the sign of the second eigenvector.

Figure 6.7: A sexual eigenvector? The face to the left might be said (intuitively) to have typically male characteristics, while the face on the right might be said to have some typically female characteristics. They are in fact the same eigenface (the second one)! The image to the left is the original, scaled, image. The image to the right is the photographic negative of the image to the left.

ingredient for a commercial package is a front-end to the algorithm that captures a face and puts it into some canonical form. Alternatively, one can use sub-blocks of the face for coding (this is what the FaceIt [6] algorithm does, which has also appeared in the popular media).

Exercise: Reproduce the figures given in this section using the data found on the class website.

6.9 A Movie Data Example

A movie is also a collection of photographs, so that we can similarly construct a "best basis" for a movie segment. In this example, the author sat in front of an inexpensive webcam and tried to move in a periodic fashion. Figure 6.8 shows some sample frames from the movie[3]. Each of the 109 frames was 120×160 pixels, and the movie is in grayscale.

To reconstruct the movie in Matlab, use the following commands:

```
load author1 %This is a matrix Y1 that is 19,200 x 109

for j=1:109
   A=reshape(Y1(:,j),120,160);
   imagesc(A);
   if j==1
      colormap(gray);
   end
   M(j)=getframe;
end
```

To replay the movie, type `movie(M)`.

Exercise: Find the best basis for the movie clip. Plot the mean image and the first five eigenvectors (as movie frames or images). Plot the movie in the best three dimensional coordinate system. Comment on what you see- for example, is the movie actually periodic? What kinds of features are being encoded in the eigenvectors?

[3]As a technical note, the webcam recorded the motion in AVI format, which was subsequently converted to MPEG and then split using software that is freely available on the internet. A script was then written to load each frame into Matlab. Subsequent versions of Matlab can now read some forms of movies directly- See the help section for `imread` and `movie`.

Figure 6.8: Four sample frames from the 109 frame movie. Each frame is 120×160.

Student Version of MATLAB