

**Face Recognition by using Eigenfaces**

**Thanh Nguyen**

**Introduction**

**Eigenfaces** is the name given to a set of [eigenvectors](http://en.wikipedia.org/wiki/Eigenvector) when they are used in the [computer vision](http://en.wikipedia.org/wiki/Computer_vision) problem of human [face recognition](http://en.wikipedia.org/wiki/Facial_recognition_system).[[1]](http://en.wikipedia.org/wiki/Eigenface#cite_note-1)The approach of using eigenfaces for [recognition](http://en.wikipedia.org/wiki/Facial_recognition_system) was developed by Sirovich and Kirby (1987) and used by [Matthew Turk](http://en.wikipedia.org/w/index.php?title=Matthew_Turk&action=edit&redlink=1) and [Alex Pentland](http://en.wikipedia.org/wiki/Alex_Pentland) in face classification.[[2]](http://en.wikipedia.org/wiki/Eigenface#cite_note-2) The eigenvectors are derived from the [covariance matrix](http://en.wikipedia.org/wiki/Covariance_matrix) of the [probability distribution](http://en.wikipedia.org/wiki/Probability_distribution) over the high-[dimensional](http://en.wikipedia.org/wiki/Dimension) [vector space](http://en.wikipedia.org/wiki/Vector_space) of face images. The eigenfaces themselves form a basis set of all images used to construct the covariance matrix. This produces dimension reduction by allowing the smaller set of basis images to represent the original training images. Classification can be achieved by comparing how faces are represented by the basis set.

**The Eigenfaces Approach Idea**

This approach to face recognition involves the following initialization operation:

* Acquire an initial set of face images (including 360 images from 40 people, 9 images for each in different angle and emotion).
* Calculate the eigenfaces from the training set, keeping only M images define the face space.
* Calculate the corresponding distribution in M-dimensional weight space for each known individual, by projecting their images onto the “face space”.
* Find the face class that minimizes the Euclidian and matching with face in the training set.

**Theory and procedure**

***Calculating eigenface***

Let the image I(x,y) be a two-dimensional N by N array of (8-bit) intensity values. The images are converted to double values in Matlab and considered as a vector of dimension N2. In experiment, a typical image of size 112x92 becomes a vector of dimension 10304 dimensional space.

The principal component analysis is used to find the vector that best account for the distribution of face images within the entire image space. These vectors define the subspace of face image which called “face space”.

Let the training set of face images be Г1, Г2, Г3 . . . ГM. The average face of set is defined by:

Each face differs from the average by the vector:

This set is very large vectors is then subject to PCA, which seek a set of M orthonormal vectors, **u**n, which best describes the distribution of the data. The *k*th vector, **u**k, is chosen such that

The vector **u**k and the scalars **λ**k are the eigenvector and eigenvalues, respectively, of the covariance

Where the matrix , and = pixel i of face n,The matrix C, however, is N2 by N2, and determining the N2 eigenvectors and eigenvalue is an in typical task of typical image size. Since the number of data points in the image space is less than the dimension of the space (M<N2), there will be only M-1, rather than N2, meaningful eigenvector. We can solve for the N2 dimensional eigenvectors in this case by first solving for eigenvector of an M by M matrix (360 x 360) matrix rather than a 10304 matrix.

PCA tells us that since we have only M image, we have only M-non-trivial eigenvector. We can solve for these eigenvectors by taking the eigenvectors of a new M x M matrix: where and find the M eigenvectors, , of . The reason of find because of the following math trick:

We can see that is an eigenvector of C. The M eigenvectors of L are finally used to form the M eigenvectors of C that form our eigenface basis:

It turns out that only M-k eigenfaces are actually needed to procedure a complete basis for the face space.

In experiment, we only use only a few eigenfaces (M’), where M’ = 20 corresponding to the vector with the highest eigenvalues and represent the most variance with the face space.

At this step, we already 20 highest eigenfaces, next step is to use there eigenface to classify the new face image.

***Using Eigenfaces to classify the new face images***

A new face image () is transformed into its eigenface component (which the projection on “face space”) by the operation:

This describes a set of point-by-point image multiplications and summations, operations performed at approximately farm on current image processing hardware.

The weights form a vector that describes the contribution of each eigenface in representing the new face image, considering the eigenfaces as a basis set for face images.

An incoming image can similarly be projected onto the face space. This will yield a vector in M dimension space. Logically, faces of the same person will map fairly closely to one another in this face space. Recognition is simply a problem of finding the closest database image, or mathematically finding the minimum Euclidian distance between a test point and a database point.

where  is a vector describing the *k*th face class. We can use the threshold to classify the new image in three case:

* Not a face image
* A face image but no matching with dataset
* A face image and matching with image in dataset

The simulation below is just to test the new face image is belonged to the dataset or not.

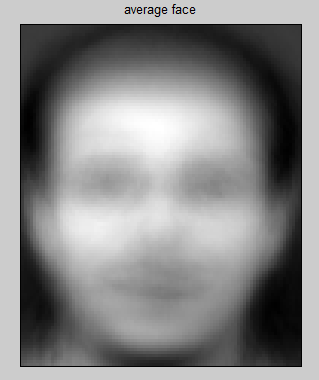
***Simulation on Matlab***

The database contains 400 images of 40 people. Each person has 10 face images with different looking directions and emotions. One of those 10 images is picked to create the new face images used for testing. The rest is considered the training set. So we have 40 new images for testing and 360 training images. The simulation was done on Matlab.

40 new images of 40 people are below:



The average face of training set:





Here is only 20 highest eigenfaces

Classifying the new faces with the training set



**Conclusions**

”Eigenfaces for Recognition” seeks to implement a system capable of efficient, simple, and accurate face recognition in a constrained environment (such as a household or an office). The system does not depend on 3-D models or intuitive knowledge of the structure of the face (eyes, nose, and mouth). Classification is instead performed using a linear combination of characteristic features (eigenfaces).

Various extensions have been made to the eigenface method such [eigenfeatures](http://en.wikipedia.org/w/index.php?title=Eigenfeatures&action=edit&redlink=1" \o "Eigenfeatures (page does not exist)). This method combines [facial metrics](http://en.wikipedia.org/w/index.php?title=Facial_metrics&action=edit&redlink=1) (measuring distance between facial features) with the eigenface representation. This method for facial recognition is less sensitive to variation in lighting and pose of the face than using eigenfaces.

**References**

[1] M.A.Turk and A.P.Pentland, ―Face Recognition Using Eigenfaces, Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition, p. 586-591, June 1991

[2] Eigenface-based facial recognition‖ by Dimitri PISSARENKO

[3] Face Recognition using Eigenfaces by Marshall Robinson, Matthew Escarra, Jon Krueger, Doug Kochelek, 2008