

**Image Processing Techniques for Face
Recognition**

University of Wisconsin Madison

ECE 533 Project

Adam Schreiner

12/21/2006

Introduction

Face recognition is a rapidly growing field today for its many uses in the fields of biometric authentication, security, and many other areas. There are many problems that exist due to the many factors that can affect the photos. When processing images one must take into account the variations in light, image quality, the person's pose and facial expressions along with others. In order to successfully be able to identify individuals correctly there must be some way to account for all these variations and be able to come up with a valid answer.

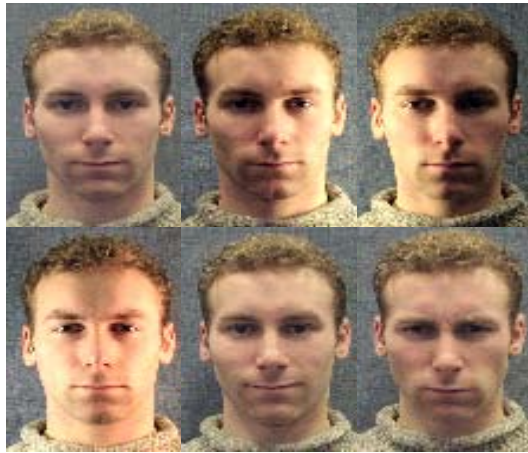


Figure 1) Differences in Lighting and Facial Expression

Approach

In order to come up with a method that will help increase the chances of correct matches I propose to apply methods we have learned this year to “preprocess” the images before they are sent into the database to be matched. This should help to remove some of the major differences that can show up in the images. In order to verify the results of this

processing I am going to implement the eigenface approach proposed by Turk and Pentland which can be found in there paper here, <http://face-rec.org/algorithms/PCA/jcn.pdf>

Process

First I implemented the calculations for the eigenfaces which I will give a brief overview of taken from Turk and Pentland [1].

- 1) Acquire an initial set of face images (the training set) $\Gamma_1, \Gamma_2, \dots, \Gamma_M$

I used the ORL database [2] available in the public domain and linked form the ECE 533 course homepage [3]

- 2) Calculate the eigenfaces from the training set, keeping the M best images and there corresponding eigenvalues to make up the face space.

In order to calculate the eigenfaces I followed the method I have outlined below.

Calculate the average image of the training set(ψ): $\psi = \sum_{n=1}^M \Gamma_n$ $\Phi_i = \Gamma_i - \Psi$

Find the difference of each face form average face(Φ): $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$

Calculate the matrix L and calculate its eigen vectors (u) $L = A^T A$ $u_i = \sum_{k=1}^M u_{ik} \Phi_k$

From which you can calculate the eigen faces: (ω) $\omega_k = u_k^T (\Gamma_l - \Phi)$

And can then from the set of weights for each image (Ω) $\Omega_k^T = [\omega_1, \omega_2, \dots, \omega_M]$

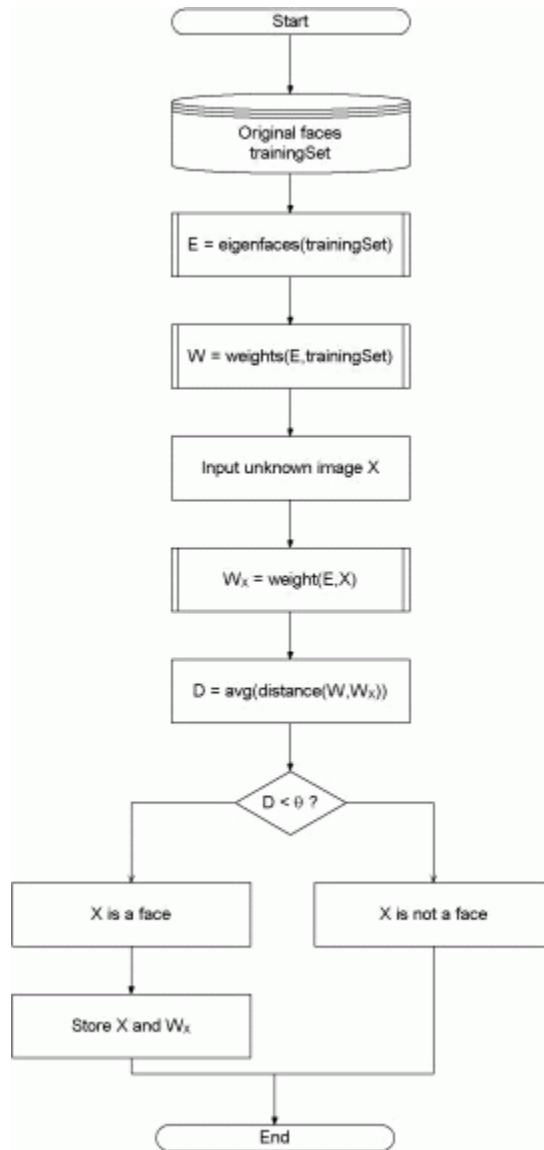


Figure 2) Face Recognition Process [4]

Once the eigenfaces are known you can take an input image and in the same way calculate its eigenfaces from the known data and use this to classify it to a known face value. I chose to use the Euclidean distance as done by Turk and Pentland to calculate the known face.

Once everything was in place I was able to implement in my software access for the user to implement up to two filters to be applied to the images before they were processed for recognition. Matlab code is attached at the end.

Results

I measured the number of correct matches found for each of fifteen different cases which can be found in the table below, each case was run on 60 images with a training set of 140 images. I also took measurements of each for both the full resolution (92x112) in this picture set and for a smaller version at 1/3 scale to see performance differences. Finally I also took measurements of the time for each to complete in order to determine which process gave the best results.

| Filter | 1/3 resolution | t | Full Resolution | t |
|-----------------------------|-------------------|---------------|--------------------|--------------|
| 3x3 gaussian | 20 | 6.9 | 28 | 19.24 |
| 5x5 gaussian | 20 | 8.14 | 22 | 19.67 |
| 3x3 avg | 16 | 8.08 | 23 | 18.64 |
| 5x5 avg | 27 | 11.033 | 20 | 21.9551 |
| 3x3 weighted avg | 18 | 10.02 | 23 | 22.5 |
| sharpen | 13 | 7.488 | 15 | 19.35 |
| sharpen2 | 10 | 6.89 | 10 | 7.34 |
| sobel1 | 12 | 10.34 | 16 | 20.99 |
| sobel2 | 14 | 10.2 | 8 | 21.6 |
| sobel | 14 | 14.61 | 8 | 25.36 |
| prewitt1 | 16 | 10.59 | 12 | 21.76 |
| prewitt2 | 11 | 9.82 | 12 | 21.22 |
| prewitt | 11 | 14.97 | 12 | 23.98 |
| 5x5 avg and 3x3 gaussian | 27 | 12.66 | 20 | 21.85 |
| Sharpen and 5x5 avg | 13 | 12.1 | 15 | 21.9 |

Figure 3) Results

Discussion

As can be seen from the above table the 5x5 averaging (blur) filter came back with not only the second best results with 27 matches but also without relatively longer processing time and being a lower quality image. In fact many of the lower quality images performed better than the full scale resolution counterparts. Using the 3x3 Gaussian filter on the full scale image achieved one additional match but at almost twice the processing time would not be an optimal solution to be implemented on a large scale.

Future Work

As can be seen my results are nowhere near perfect yet with my best results coming in with almost 50% matches. I would like to improve my code for image verification as well as clean up the code in order to improve performance.

- 1) M. Turk, A. Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991, pp. 71-86
- 2)ECE 533 course homepage, <http://homepages.cae.wisc.edu/~ece533/>
- 3) F. Samaria and A. Harter , "Parameterisation of a stochastic model for human face identification", 2nd IEEE Workshop on Applications of Computer Vision December 1994, Sarasota (Florida).
- 4) Dimitri PISSARENKO, Eigenface-based facial recognition, <http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html>, February 13, 2003
- 5) Matthew Dailey, UCSD Artificial Intelligence Laboratory, <http://ai.ucsd.edu/>
- 6) R Gonzalez and R Woods, Digital Image Processing, Prentice Hall, 2002