

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) $\Omega_k^T = [\omega_1, \omega_2, \dots, \omega_M]$