

Face Detection using Template Matching

Deepesh Jain
Husrev Tolga Ilhan
Subbu Meiyappan
(Group 12)

EE368
Spring 2003
Stanford University
Demo: 05/26/03

Table of Contents

Face Detection using Template Matching..... 1

1. Features..... 3
2. Assumptions..... 3
3. Architecture..... 3
4. Investigated Methods..... 3
5. Theory..... 4
 - 5.1. Color Segmentation 4
 - 5.2. Template Matching..... 5
 - 5.3. Principal Component Analysis (PCA)..... 6
 - 5.4. Reconstruction 7
6. Results..... 8
7. Conclusion 10
8. Work Distribution 10
9. References:..... 10

Face Detection using Template Matching

1. FEATURES

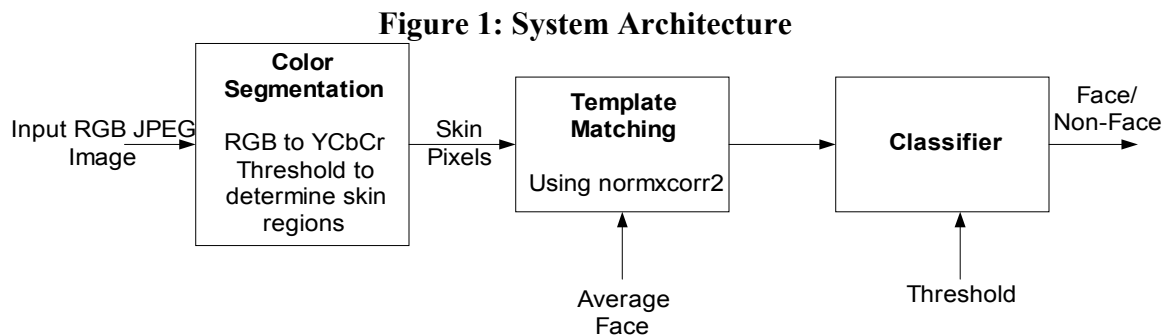
- Color segmentation in YCbCr and HSV space
- Dual detection procedure using Template Matching and Principal Component Analysis

2. ASSUMPTIONS

- The image set can contain many faces and non-faces
- The image set contains only frontal view faces
- The image set can have a few faces that are occluded (partly hidden)
- The image resolution and size are the same as the original image (1856x1392 pixels)

3. ARCHITECTURE

The architecture of the system that was implemented is as shown in Figure 1.



4. INVESTIGATED METHODS

Several methods for face detection were studied and investigated. We document a few methods which we explored, for face detection

- **Template Matching** : An average face of a select set of faces from the entire image set was obtained from the ground truth data. This average face was then cross correlated with the skin segmented image. Using a threshold detector, it was determined if it was a face or not. Template matching was tested at various levels:
 - Average face template
 - Average Left Eye/Right Eye templates
 - Average eyes template
 - Average mouth template
 - Average nose template

Of all these templates only the face template seem to give better results and hence that was retained.

- **EigenFaces**: From a set of faces from the ground truth data, eigen images were obtained using the Sirovich and Kirby method. A set of these eigen images

were then used as the basis images to reconstruct the images. The mean squared error between the reconstructed face and the original face was used as a metric for determining whether the input image belonged to the face class.

Other classification methods that were investigated include Mahalanobis distance metrics. Eigen images of the nonfaces were also computed to determine if the reconstructed image was closer to the face space or non-face space. But this approach was eventually dropped.

Eigeneyes based approach was also investigated as an alternative for detecting occluded images. The eigenface basis was also used to investigate gender classification. Because of face occlusion, these methods did not work very well and hence were dropped.

- Wavelets and Neural Nets: The use of wavelets for multiresolution analysis and artificial neural networks (in particular, the Learning Vector Quantization) for classification was investigated.

5. THEORY

Template matching and Eigenimage methods were implemented for the purpose of image detection. This section describes the theoretical and implementation aspects of these two methods.

5.1. COLOR SEGMENTATION

We have segmented out the skin portions of the image by filtering out the pixels with non-skin color. The skin color is detected by utilizing color distribution of the skin pixels in (**cR**, **cB**, **hue**) space. This space is a combination of the YcBcR and Hue color spaces. A pixel is labeled as a skin pixel if it satisfies the following thresholds:

$$140 \leq cR \leq 160$$

$$100 \leq cB \leq 150$$

$$0.1 \leq hue \leq 0.9$$

The boundaries are determined empirically. Satisfying these conditions is not enough to completely segment the face pixels. A face has to cover certain area, and should have certain height and size. We assumed that the area of a face should be more than 800 pixels and its height and width should range between [80, 160] pixels.

Figure 2 shows the results of our color segmentation algorithm for one of the training data sets. It is not possible to separate faces if they are very close to each other (especially if one face is occluded by another.) In such cases we used erosion/dilation repetitively. Still, for some cases each frame returned by the segmentation algorithm included more than one face.



Figure 2: Skin Color Segmentation

5.2. TEMPLATE MATCHING

We first tried the template matching method. As it is seen in figure 3, our template is the average of the faces in the 7 training set.

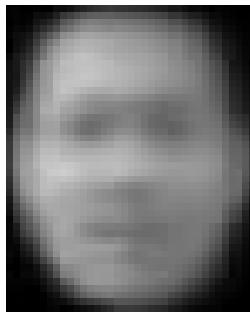


Figure 3: Average Face

For each frame out of the color segmentation block, we formed an image pyramid with 6 scales (figure 4(a).) At each scale, the cross correlation of the template with the image is calculated to detect the face (figure 4(b)). The frame at the last scale is only 40% of the frame at the first scale.

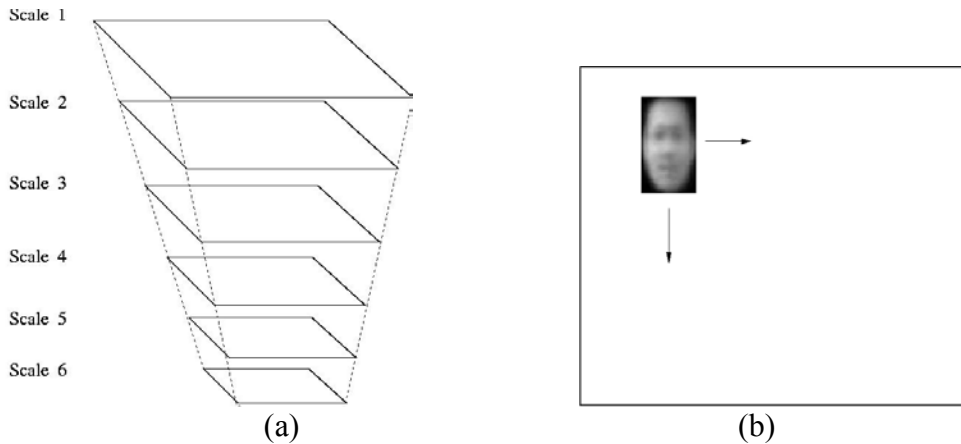


Figure 4: (a) Image Pyramid (b) Template Matching

We assumed that the center of the detected face is wherever the cross-correlation is maximum. Of course, there is a threshold used for the correlation value to be sure that what is detected is a face.

In case there are more than one face in a given frame, we do not stop after a face detected. We first remove the currently detected face then reprocess the frame until no more face remained to be detected. Figure 5 illustrates this process

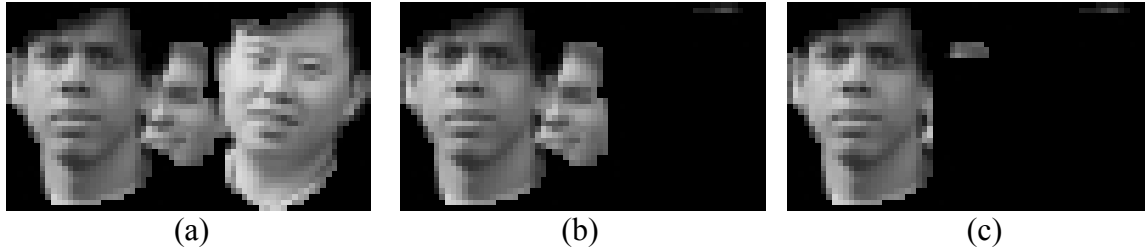


Figure 5: (a) Original Image (b) 1st. face detected (c) 2nd face detected

Surprisingly, this method performed better than we expected. The results can be seen in the Results section below.

5.3. PRINCIPAL COMPONENT ANALYSIS (PCA)

There is vast literature on the use of principal component analysis for face recognition. The method discussed in the EE368 (due to Sirovich Kirby) in class was used in this project to obtain the eigenimages.

PCA computes the basis of a space which is represented by its training vectors. The basis vectors computed by PCA are in the direction of the largest variance of the training vectors. These basis vectors are computed by solution of an eigen value problem, and as such the basis vectors are eigenvectors. These eigenvectors are defined in the image space. They can be viewed as images and indeed look like faces. Hence they are usually referred to as eigenfaces.

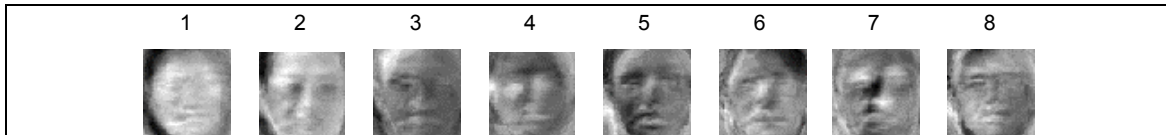


Figure 6: First Few EigenImages

The first eigenface is the average face, while the rest of the eigenfaces represent variations from this average face. The first eigenface is a good face filter: each face multiplied pixel by pixel (inner product) with this average face yields a number close to one — with non-face images the inner product is far less than one. The direction of the largest variation (from the average) of the training vectors is described by the second eigenface. The direction of the second largest variation (from the average) of the training vectors is described by the third eigenface, and so on.

Each eigenface can be viewed as a feature. When a particular face is projected onto the face space, its vector (made up of its weight values with respect to each eigenface) into the face space describes the importance of each of those features in the face. For convenience, the weight vector is normalised. Since the image developed in the face space is indeed a face, the weight of the first eigenface is very high, almost equal to unity. (This useful property may be used to test images for face-like qualities). The value of the weights decreases as the number of the eigenface increases. This is in conformity with the definition of eigenfaces. In fact, PCA finds the direction of largest variations. The first eigenface accounts for the maximal variation, the second one accounts for the second maximal variation, etc.

5.4. RECONSTRUCTION

Once the eigenfaces have been computed, each face in the image space can be viewed in the face space. The transformation from image space to face space is fairly simple.

Let

- U be the matrix of the first eigenfaces, where the first column is the first eigenface and so forth,
- e_I be a face in the image space, and
- e_F be the same face in the face space.

$$e_F = e_I * U^T \quad (3.1)$$

This is a many-to-one transformation, since the dimensionality of the image space is far larger than the dimensionality of face space. Thus, the transformation introduces an error which can be seen by looking at a reconstructed face. The reconstruction is performed by taking the inverse transformation of 3.1:

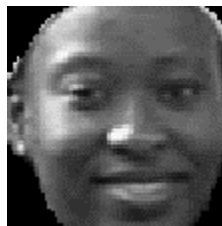
$$e_I = e_F * U$$

As an example, Figure 3 shows a face along with its reconstruction. Note that the reconstruction error with 30 of the 48 possible eigenimages is quite minimal.

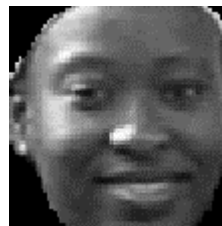
Figure 7: Original and Reconstructed

MSE = 1.19e-022

original



reconstructed



The magnitude of the difference between the original face and its reconstruction, called the reconstruction error, is easily calculable. Let e'_1 be the reconstructed face of the original face e_1 , and ε , the reconstruction error:

$$e'_1 = U^T * U * e_1 \text{ and } \varepsilon = |e_1 - e'_1|$$

The mean squared error between the original image and the reconstructed image was used as a metric to classify the image as a face or nonface.

6. RESULTS

After extensive implementation, the idea of eigenimages was dropped because it did not work well with occluded faces. This can also be explained analytically by that these occluded/partial images were not part of the original eigenimages. So, the mean square error for the occluded images are rather significant. Hence, it was decided to fine tune the template matching approach to detect all the faces. For a simplistic method, template matching works quite well.

Table 1 summarizes the results from running the faceDetect program on the training set using the template matching approach. Note that these run times were on a 1GHz Pentium III PC running Windows XP with 512MB of RAM.

Training Data	Final Score	Detect Score	Number of Hits	Number of Repeats	Number of false positives	Distance to centroid	Elapsed CPU time
Training_1.jpg	21	21	21	0	0	12.10	163.87
Training_2.jpg	20	20	23	1	2	16.61	172.76
Training_3.jpg	23	23	25	0	2	8.84	161.54
Training_4.jpg	21	21	24	1	2	15.87	133.66
Training_5.jpg	23	23	23	0	0	11.91	146.11
Training_6.jpg	23	23	24	0	1	9.46	147.51
Training_7.jpg	21	21	22	1	0	17.55	198.78

Table 1: Template Matching Results

Figure 8 and Figure 9 show the result of the algorithm on a sample of the training sets.



Figure 8: Results on Training_1.jpg

Results for Training_7.jpg



Figure 9: Results on Training_7.jpg
(The yellow cross denotes repeated hits on the same face)

7. CONCLUSION

- Template matching was used successfully to detect faces in color images
- Using template matching, occluded faces were also detected
- Eigen-face-image approach does not work very well with occluded faces
- The accuracy of the implemented template matching was over 92%

8. WORK DISTRIBUTION

- Literature Survey : All
- Investigation on the use of wavelets and multi resolution analysis : Husrev Tolga Ilhan
- Color segmentation : Deepesh Jain
- Template matching : Deepesh Jain and Subbu Meiyappan
- Principal Component Analysis : Husrev Tolga Ilhan and Subbu Meiyappan
- Testing : All
- Documentation : All

9. REFERENCES:

1. M. KIRBY, AND L. SIROVICH, *APPLICATION OF THE KARHUNEN-LOÈVE PROCEDURE for the characterization of human faces*, IEEE transactions on pattern analysis and machine intelligence, Vol. 12, No. 1, (1990).
2. EE 368 Spring 2002-2003 class notes <http://www.stanford.edu/class/ee368/>
3. Sami Romdhani, "Face Detection," Master's Thesis, <http://graphics.informatik.uni-freiburg.de/people/romdhani/>
4. R. Gonzales and R. E. Woods, "Digital Image Processing," Prentice Hall, Inc, New Jersey (2002)
5. Anil K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall, Inc., New Jersey (1989)
6. M. A. Turk and A. P. Pentland, *Face recognition using eigenfaces*, Proc. of the IEEE on Computer Soc. Conf. (1991).