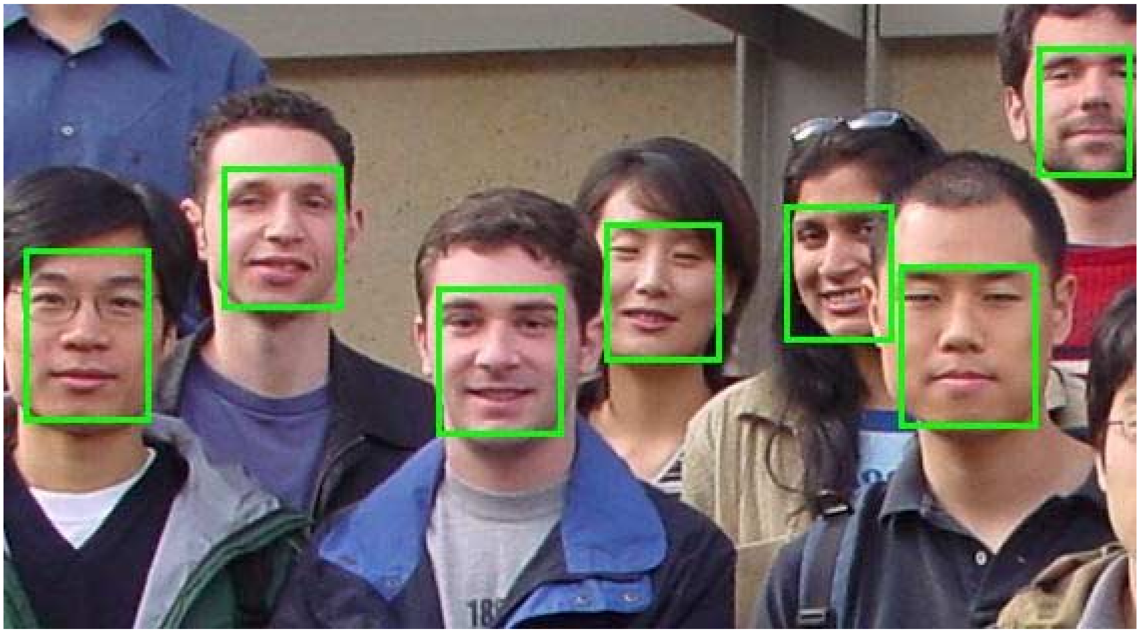

Face Detection on Similar Color Photographs

Scott Leahy

EE368: Digital Image Processing
Professor: Bernd Girod
Stanford University

Spring 2003

Table of Contents

1.0 Introduction

The basic skill of detecting faces in an image is something that humans often take for granted. Creating a computer program to perform the same task turns out to be a difficult problem for which more effective and more efficient algorithms continue to surface. In this project, the author attempts to combine some of these algorithms with the basics that were learned in EE368 to create a face detection algorithm for a specific set of images.

Seven images were provided as a way of developing and testing the class' algorithms. Each of these images had many similarities to each other and presumably to the final test image. The images were full color photographs of the Spring EE368 class of 2003. All of the pictures were taken in the same location, but with different arrangements of the students and professors. By taking advantage of these similarities, an effective algorithm has been developed that will hopefully be successful when performed on the final test image.

2.0 Face Detection Mechanism

The basic form of this algorithm follows a process that was very common among former EE368 students. The following two steps are performed:

1 – Skin Detection – Since the training set and the final image are all full color images, the separation of skin pixels from non-skin pixels can be accomplished quite effectively.

2 – Template Matching – By running only the skin pixels through a template matching algorithm, the faces can be separated from other visible skin such as arms or legs.

Using these techniques, the faces in the training images were recognized quite effectively. Although better approaches exist for more general circumstances, this approach works well for this particular situation.

2.1 Skin Detection

To accomplish the task of separating skin pixels from non-skin pixels, a concept was borrowed from digital communications. The Maximum Apriori Probability (MAP) detector in digital communications is based on Bayes' Rule, which states:

$$P(a|b) = P(a \cap b) / P(b)$$

The goal of the MAP detector is to maximize the probability of guessing the input based on the output. The probability of the decision $m' = m_i$ being correct, given the channel output vector $\mathbf{y} = \mathbf{v}$, is [1]

$$P_c(m' = m_i, \mathbf{y} = \mathbf{v}) = P_{m|\mathbf{y}}(m_i|\mathbf{v}) = P_{x|\mathbf{y}}(i|\mathbf{v})$$

In other words, the probability of making a correct decision is maximized by finding the most likely input based on the output vector. Often, however, this information is not explicitly known in a system. Rather than having a probability density function (PDF) of the input given the output, it is much more common to know the probability of the output given the input. Noting that the sum of the probabilities of all of the outputs equals one and using Bayes' Rule as described above, the MAP decision rule can be restated as follows [1]:

$$\textit{Decide input} = m_i \ \ \textit{if} \ \ p_{\mathbf{y}|x}(\mathbf{v}|i) * p_x(i) \geq p_{\mathbf{y}|x}(\mathbf{v}|j) * px(j) \ \ \cancel{\forall} \ \ j \neq i$$

In other words, by knowing the PDF of the output given the input as well as the probability of the input itself, the probability of error can be minimized (assuming the outputs are uncorrelated with one another).

In the skin detection application, the input is simply "skin" or "non-skin". The output can be any set of coordinates in RGB, HSV, or any other color space. The probability of the inputs, (skin vs. non-skin,) can roughly be determined using the training images. The final image is assumed to have a similar probability distribution of skin vs. non-skin values. The only remaining variable that is needed to perform MAP detection is a PDF of the color space given the input.

To accomplish this, histograms were created in HSV space using all of the pixels in all of the training images. Histograms were created in each of the three coordinates of HSV space using each of the two inputs (skin and non-skin). Figure 1 below shows the results of these histograms after being normalized based on the probability of the two inputs.

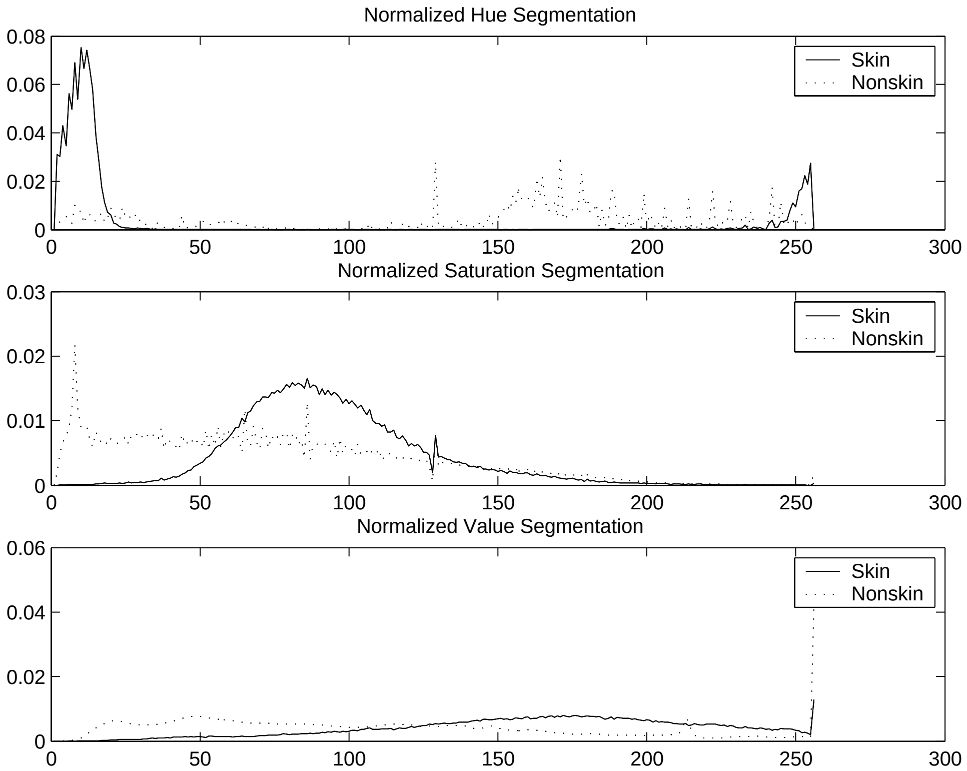

Figure 1: Histogram of Skin vs. Non-skin in HSV Space

As the histograms show, there is a fairly distinct separation between skin and non-skin in the hue and saturation coordinates with less separation in the value coordinate. Using only the first two coordinates, a joint PDF was created that could be used in the decision process. Each pixel from the test image was passed through the resulting MAP detector, and the following "skin" mask was created.

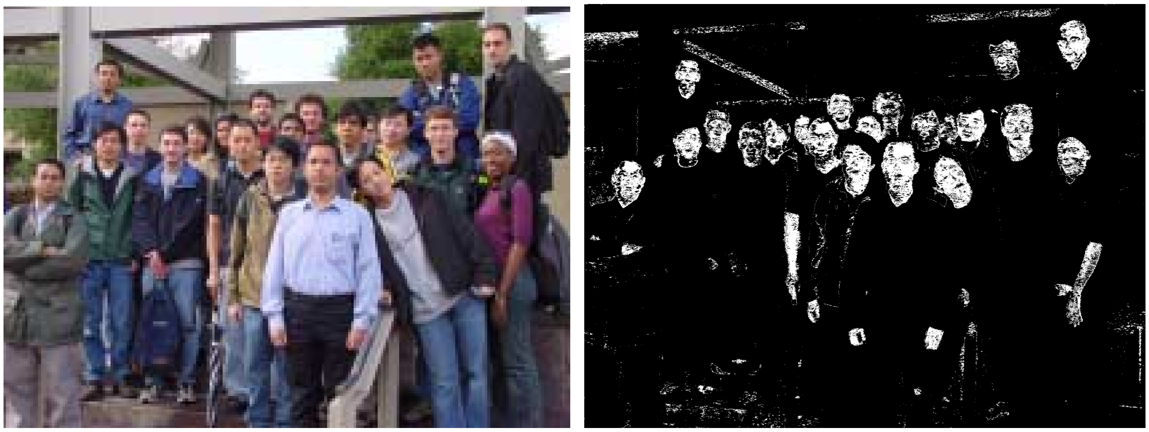

Figure 2: Result of using Hue and Saturation for MAP Detector

While the resulting skin mask was certainly a nice approximation, there were still some annoying artifacts. For example, the undersides of the beams in the background were still visible as were some pixels in the clothing of the people in the image. In an attempt to improve on this approximation, the third remaining coordinate, value, was added to create a three-dimensional joint PDF of the entire color space. The final implementation of the skin detector used this PDF to perform the skin vs. non-skin approximation. Figure 3 below shows an example of the final results of this algorithm on one of the training images.

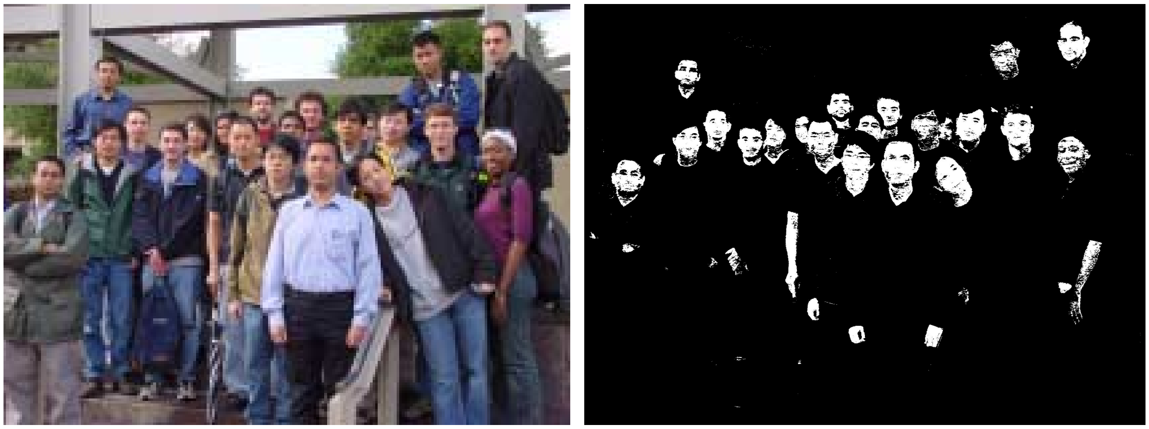

Figure 3: Result of using Hue, Saturation, and Red for MAP Detector

The resulting skin mask appears to contain only skin pixels, but it also appears to have some holes. To resolve this problem, some binary morphological processes were performed. The image was first dilated in order to try to connect the pixels in each individual face. A “filling” step was then performed to fill in any holes such as in the eyes and under the chin. Finally, the image was eroded to remove the extra pixels that were selected around the edges of the faces during the dilation step. Figure 4 below shows an example of the final step of this procedure.

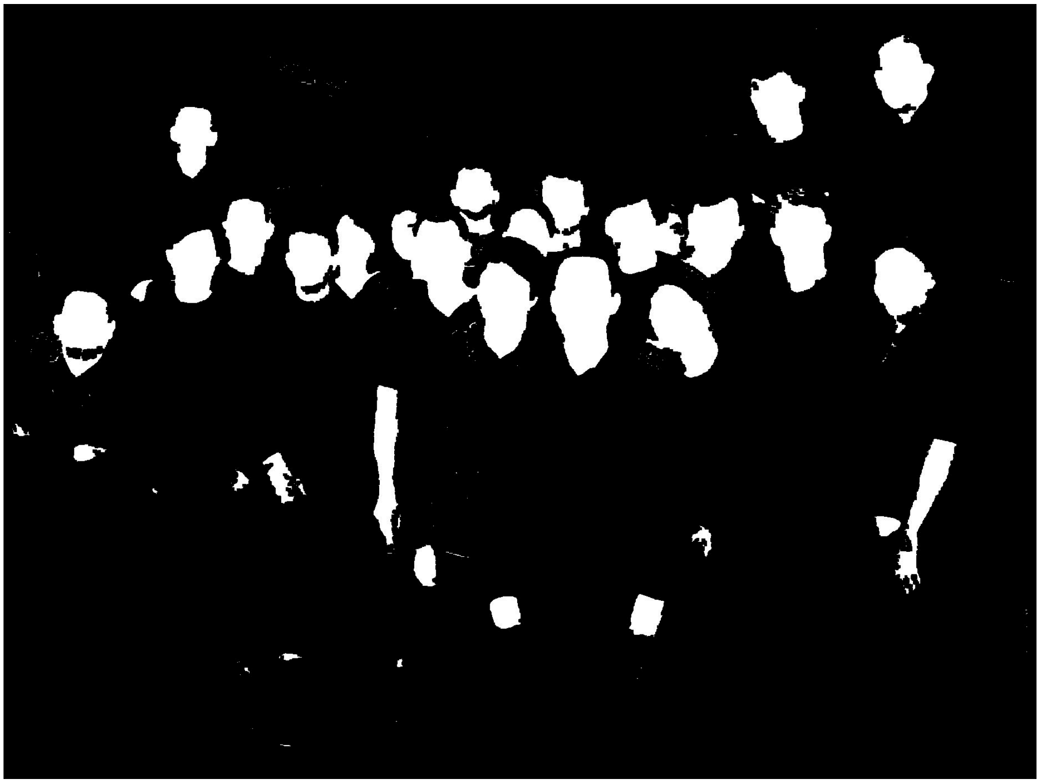

Figure 4: Example of Result of Skin Detection Step

2.2 Template Matching

The resulting masks from the skin detection step removed approximately 90-95% of the pixels from the image as being face pixels. This greatly reduced the computational time required for the next step, template matching.

The basic idea of template matching is to create an approximation of the target object in the image, then to scan through the image looking for the best matches to that template. Finding the best matches to the template can be done in one of two ways: 1) finding blocks in the image with the lowest mean square error to the template, or 2) finding the blocks in the image with the highest correlation to the template. (It can be shown that these two methods are equivalent as long as the mean-square values of the test blocks are normalized.) The only caveat to this method is that the mean value of the test blocks must be removed before performing the process in order to avoid biasing toward the brighter areas of the image. [2]

While the concepts behind template matching are fairly easy to understand, creating an implementation for template matching can be something of an art. There are many ways of creating templates, and the threshold that is chosen for which parts of the image are to be considered matches and which are not can have drastic effects on the effectiveness of the algorithm.

For this project, two different templates were tested. (See figure 5 below.) The first template was created by manually selecting about 30 faces out of the original image then creating a template from the average intensity of the faces. Originally this was not expected to be a very effective template because there was not a great amount of contrast in the image, and the image only faintly resembles an actual face. The second template was found on a website dedicated to face detection algorithms, created and supported by Robert Frischholz. The image is “a typical averaged face” according to the site. [3] This image was quite a bit more crisp than the image created from the average of the training set’s faces as can be seen below.

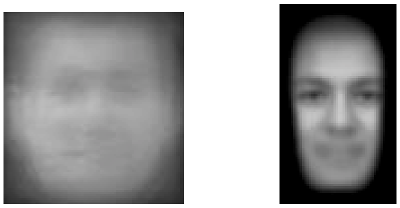

Figure 5: Two Templates Used in Face Detection

Using these two templates, the thresholds for deciding face vs. non-face were tweaked until the best results were obtained. Unfortunately, although both templates did a decent job (60-80% success rates) they also made quite a few errors in declaring objects such as hands and necks as faces.

In an attempt to improve on this, the two templates were cropped so that the resulting images only included the eyes and nose of the templates. The resulting templates are shown in figure 6 below.

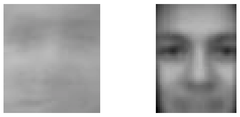

Figure 6: Cropped Templates Used in Second Trial

Not only did both of these templates perform much better than the first two, but they also required significantly less processing time since many fewer multiplications needed to be performed. In the end it was found that the template that was created from an average of the faces in the training images gave the highest success rates. An example of the result of this step is shown in Figure 7 below.

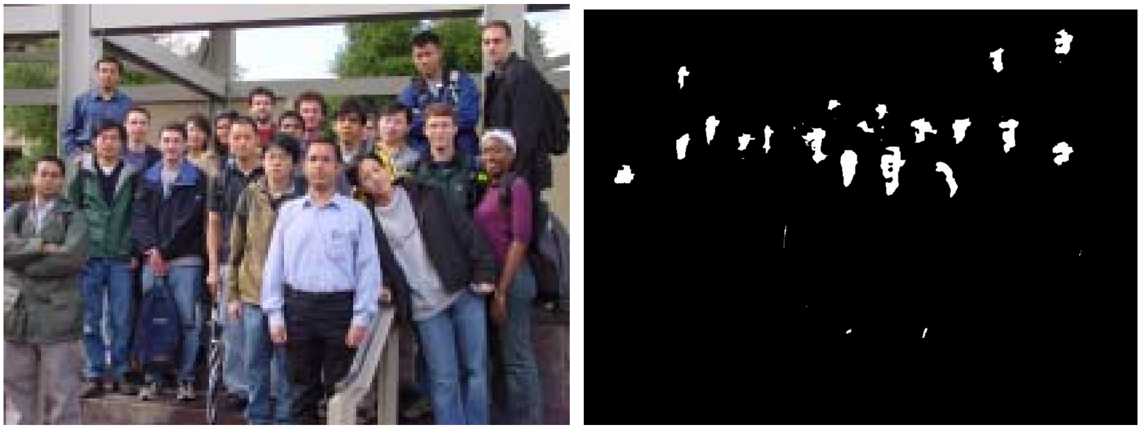

Figure 7: Result of Template Matching Step

Once this final mask was created, some simple steps were taken to locate the regions that were to be declared faces. Each white region in the image was identified, and if a region turned out to be bigger in area than a given threshold, it was declared a face. The coordinates of the centers of the "face" regions were returned as the results of the face detection algorithm.

2.3 Unsuccessful Methods

An additional approach to face detection for this application is worth mentioning although its effectiveness did not turn out to be as good as the template matching approach. Given that the skin detection step was so successful (see figure 4), it was hypothesized that faces might be able to be detected using only morphological processing. Since faces are generally round in shape, an approach was created that involved eroding the image by a disk whose diameter was chosen to be slightly smaller than the width of the smallest face in the image.

The difficult part of this method was in finding the smallest face in the image, not knowing which white blocks in the image were faces and which were not. The approach that was taken was to approximate the smallest face by finding the closest white region to the upper left-hand corner of the image that had a significant enough area to be considered a face. Generally, the face closest to that corner ended up being in the back row of the photograph, and therefore had the one of the smallest areas of all of the faces. This approximation was crude, but it was not necessary to find the absolute smallest face in the image. An approximation was all that was desired.

Using the face in the upper left-hand corner of the image as a basis, the width of the face was measured, and a structuring element was created as a disk with a diameter about 80% as wide as the face. An erosion step was then performed using this structuring element.

The results of this method did not turn out to be as good as with the template matching method. The results became even worse on images where the closest face to the upper left-hand corner did not happen to be the smallest face, or worse yet, where two faces were adjacent to one another in the upper left such that they were considered one large region. Given these poor results, this method was subsequently discarded.

3.0 Results and Conclusions

Table 1 below shows the results of the final algorithm as performed on the seven training images.

Table 1: Results of the Face Detection Algorithm

Training Image Number	Total Score	Number of Hits	False Detections	Maximum Score	Time Required (seconds)
1	19	19	0	21	68
2	21	22	1	24	60
3	19	23	4	25	60
4	18	19	1	24	58
5	22	22	0	24	60
6	22	23	1	24	59
7	20	21	1	23	68

Note that as a bonus for this project, if the female faces could also be picked out of the image, extra points would be awarded. Due to the time constraints on this project, no algorithms were developed for this purpose.

The final results show that this algorithm has approximately an 85% success rate based on the method by which points were awarded.

4.0 References

1. Cioffi, John. EE379 Course Reader, January 2002. http://www.stanford.edu/class/ee379a/reader.html

2. Girod, Bernd, EE368: Digital Image Processing, class lecture notes, April 28, 2003.

3. "The Face Detection Homepage", Robert Frischholz, http://caslab.bu.edu/course/cs585/P2/artdodge/average_face_clipped.jpg.