EE368 Digital Image Processing Project - Automatic Face Detection Using Color Based Segmentation and Template/Energy Thresholding

Michael Padilla and Zihong Fan
Group 16
Department of Electrical Engineering
EE368 - Dr. B. Girod, Spring 2002-2003
Stanford University
Email: mtp@stanford.edu, zihongf@stanford.edu

I. Introduction

The purpose of this project has been to try to replicate on a computer that which human beings are able to do effortlessly every moment of their lives, detect the presence or absence of faces in their field of vision. While it is something that to a layman appears trivial, to implement the necessary steps leading to the successful execution of this in an algorithm is difficult and still an unsolved problem in computer vision.

In EE368 we have been given the task of using a collection of seven digital images to train and develop a system for doing just this in a competitive format. The only real limitation is that it run under seven minutes for a single file. In deriving a method of our own, we initially began by reviewing various articles on the topic as well the material covered in lecture. We explored the possibility of using some of the methods that have been explored by researchers thus far, such as neural networks, statistical methods, machine learning algorithms (SVM, FLD), PLC (such as Eigenfaces and the concept of a ”face space”), as well as a newer methodology called Maximum Rejection Classification (MRC). We initially attempted to devise a system that linked Eigenface based front-end with a neural network based back-end, but the neural network machinery proved rather difficult to train and develop in a manner that allowed us to understand the inner workings of our system. We were unsuccessful in being able to generalize the training data to unseen images and were prevented by the nature of the neural network to really have a grasp of the particular shortcomings of our system and what could be done to improve it. Hence we decided to abandon that approach and pursue a method based on color segmentation followed by template/energy matching. This system has been shown to be reasonably fast, taking on the average of 80 to 120 seconds to run, depending on the internal downsampling rate applied to the input image and various other parameters that can be adjusted. With the final parameter values that we decided on, it runs for approximately 100 seconds on a Dell 1.8Mhz Pentium IV laptop. Performance accuracy was found to range from approximately 85% to 100%.

The next few sections briefly outline the system that we developed. The system is a simple application of a color based segmentation scheme that takes advantages of patterns developed in the HSV, YCrCb, and RGB color spaces followed in series by a matched filter/template matching system.

Fig. 1. Example Input Training/Testing Image

An example image that defined the space of our task is given in figure 1.

II. Color Based Segmentation

Assuming that a person framed in any random photograph is not an attendee at the Renaissance Fair or Mardi Gras, it can be assumed that the face is not white, green, red, or any unnatural color of that nature. While different ethnic groups have different levels of melanin and pigmentation, the range of colors that human facial skin takes on is clearly a subspace of the total color space. With the assumption of a typical photographic scenario, it would be clearly wise to take advantage of face-color correlations to limit our face search to areas of an input image that have at least the correct color components.

In pursuing this goal, we looked at three color spaces that have been reported to be useful in the literature, HSV and YCrCb spaces, as well as the more commonly seen RGB space. Below we will briefly describe what we found and how that knowledge was used in our system. The result of this study is the construction of hyperplanes in the various

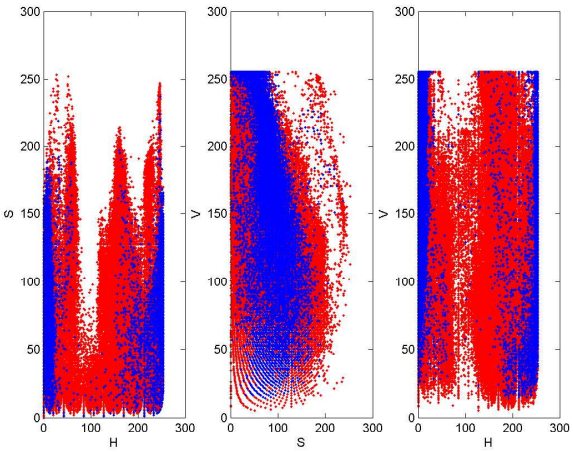

Fig. 2. H vs. S vs.V plots for face (blue) and non-face (red) pixels

Fig. 3. Remaining pixels after applying the HSV segmentation rule

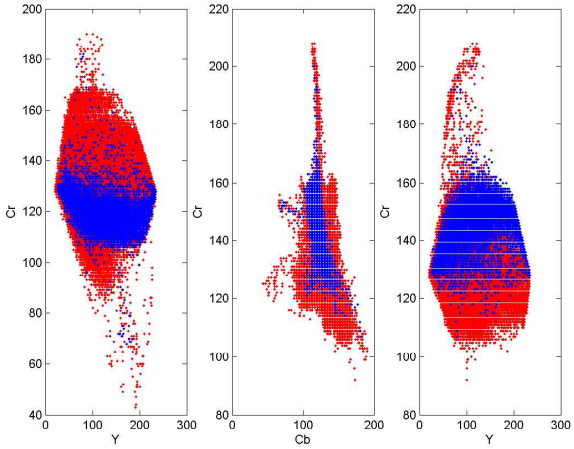

Fig. 4. Y vs. Cb vs. Cr plots for face (blue) and non-face (red) pixels

color spaces that may be used to separate colors. While elegant techniques like FLD and SVD, etc. may be used to optimally construct the hyperplanes, we built ours more ad hoc by varying the parameters of the separating lines and planes that we eventually used.

A. HSV Color Space

While RGB may be the most commonly used basis for color descriptions, it has the negative aspect that each of the coordinates (red, green, and blue) is subject to luminance effects from the lighting intensity of the environment, an aspect which does not necessarily provide relevant information about whether a particular image ”patch” is skin or not skin. The HSV color space, however, is much more intuitive and provides color information in a manner more in line how humans think of colors and how artists typically mix colors. ”Hue” describes the basic pure color of the image, ”saturation” gives the manner by which this pure color (hue) is diluted by white light, and ”Value” provides an achromatic notion of the intensity of the color. It is the first two, H and S, that will provide us with useful discriminating information regarding skin.

Using the reference images (truth images) provided by the teaching staff, we were able to plot the H,S, and V values for face and non-face pixels and try to detect any useful trends. The results of this may be viewed in figure 2. From those results it is seen that the H values tend to occupy very narrow ranges towards both the bottom and top of its possible values. This is the most noticeable trend and was used by us to derive the following rule used in our face skin detection block:

$$19 < H < 240 \Rightarrow \text{ Not Skin},$$

and otherwise we assume that it is skin. By applying a mask based on this rule to our sample image in figure 1, we have the remaining pixels seen in figure 3.

B. YCbCr Color Space

Similarly, we analyzed the YCbCr color space for any trends that we could take advantage of to remove areas that are likely to not be skin. Relevant plots may be viewed in 4.

After experimenting with various thresholds, we found that the best results were found by using the following rule:

$$102 < Cb < 128 \Rightarrow \text{Skin},$$

and otherwise assume that it is NOT skin and may be removed from further consideration. To see how our image looks after additionally applying the YCbCr rule, please refer to figure 5.

C. RGB Color Space

Let’s not be too hard on our good friend the RGB color space...she still has some useful things to offer us to take advantage of in our project. While RGB doesn’t decouple the effects of luminance, a drawback that we noted earlier, it is still able to perhaps allow us to remove certain colors that are clearly out of the range of what normal skin color is. Please refer to figure 6.

From studying and experimenting with various thresholds in RGB space, we found that the following rule worked well in removing some unnecessary pixels:

$$0.836G - 14 < B < 0.836G + 44 \Rightarrow \text{ Skin}$$

and

Fig. 5. Remaining pixels after applying the YCbCr segmentation rule

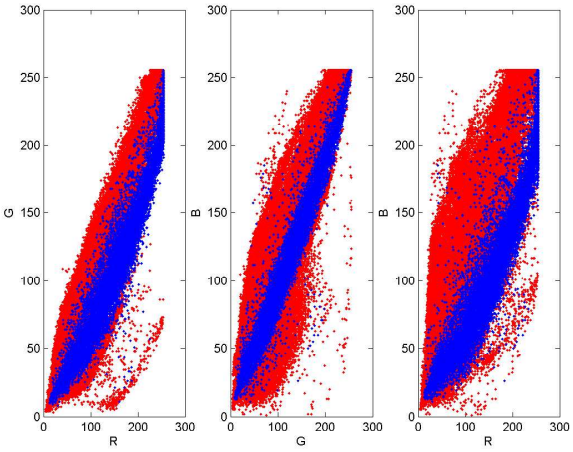

Fig. 6. R vs. G vs. B plots for face (blue) and non-face (red) pixels

$$0.79G - 67 < B < 0.78G + 42 \Rightarrow \text{ Skin},$$

with other pixels being labelled as non-face and removed. The effects of applying these two rules may be seen in figures 7 and 8.

III. Lower Plane Masking

While in general it would destroy the generality of a detector, in our case we believe that its reasonable to take advantage of a priori knowledge of where faces are most likely to be and not be to remove ”noise”. We observed that in the training images that no faces ever appeared in the lower third of the image field. With very high probability it is likely that the scenarios where our system will be used (i.e. the testing images) that the same will be true since we know that the conditions in which the pictures were taken are identical. Hence, we removed the lower portion of the image from consideration to remove the possibility of false alarms originating from this region. The additional application of this step resulted in figure 9.

Fig. 7. Remaining pixels after applying the first RGB segmentation rule

Fig. 8. Remaining pixels after applying the second RGB segmentation rule

Fig. 9. Remaining pixels after masking the lower image field

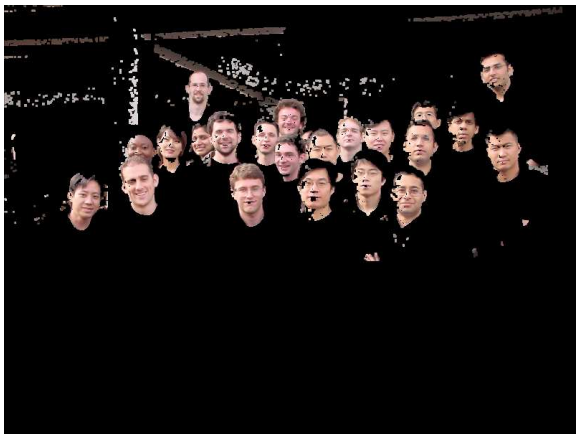

Fig. 10. Remaining pixels after applying the open morphological operator

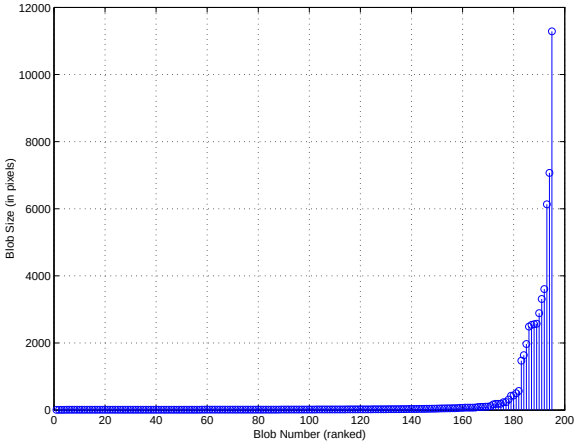

Fig. 11. Plot of ranked blob sizes

IV. Morphological Processing

A. Applying the Open Operation

At this stage in the flow or our detector (figure 9) we have successfully removed the vast majority of the original pixels from consideration, but we still see little specs throughout the masked image. Because we will subsequently send the image through a matched filter and the specs will be averaged out of consideration and hence could be left in and just ignored, it is preferable to remove them now in order to speed future processing (i.e. the matched filter needn't perform any wasteful calculations at these pixels). Hence the open (erode $\rightarrow$ dilate) operation was performed using a 3x3 window of all 1s. The result of applying this additional step is in figure 10.
It is seen that the open operation has resulted in there being a huge reduction in the number of small "noisy" specs.

Fig. 12. Final pre-processed image after small blob removal and grayscale transformation

B. Removal of Small Blobs and Grayscale Transformation

By "blobs", we simply mean the connected groups of pixels that remain at this stage. Here we may apply a little additional knowledge about the way the picture was taken...we know that the subjects in the photos were standing relatively closely to one another and hence should have head sizes (measured by number of pixels) that are relatively similar. The largest blobs should be these heads and blobs considerably smaller than the larger blobs may be safely assumed to be more "noise". In the particular sample image that we've been looking at, the sizes of the blobs from figure 10 were measured and ranked. The ranked sizes of the 195 remaining blobs is seen in figure 11.

By removing blobs that are below a given threshold size we can remove even more additional noise. After experimenting with the given image studied in this report as well as the other provided images, we found that a pixel size of 200 was a good threshold value. Hence our blob size rule is:

$$\text{Blob Size } < 200 \Rightarrow \text{ Non-face Blob,}$$

and hence such blobs may be removed. Finally, we found that after this stage in our processing that all the color information that could be used within the level of sophistication feasible for this project had been and that subsequent stages could be done in grayscale without any performance degradation, but with the additional benefit of a faster system that need only operate in one of the original three dimensions. Hence we now transform our image to grayscale. *This provides us with our final pre-processed image, which may be seen in figure* 12.

It is important to note at this point one the main problem that we faced in this project. Note that the faces are retained at this stage in the processing, but unfortunately we have been unable to resolve them into separate blobs. Were the subjects standing with sufficient separation to do allow this, we could do almost all of our necessary face detection just by working with blobs and their size statistics, etc. However, because the students in the photos are in very close clusters, multiple faces have been grouped in single blobs. This leads to complications that the template matching methodology (in our case at least) is unable to cleanly resolve in some situations.

V. Matched Filtering (Template Matching)

A. Template Design

The first task in doing template matching is to determine what template to use. Intuitively, it seemed reasonable to us that the best template to use would be one derived by somehow averaging the some images of the students in the training

Fig. 13. Selected and processed (scaled and alligned) faces for template construction

Fig. 14. Selected and processed faces after histogram normalization

images that would likely be in the testing images. We would like to find a good subset of the faces found in the training images that are clear, straight, and representative of typical lighting/environmental conditions. It is also important that these images be properly aligned and scaled with respect to one another. To this end, we spent considerable time manually segmenting, selecting, and aligning face photos. In the end we chose 30 face images, which may be seen in figure 13.

In order to have the template reflect the shape of the faces it is trying to detect, rather than their particular coloring, etc. we applied histogram equalization to each image and removed the means. This resulted in figure 14.

Our final template is a result of adding together the 30 face images in figure 14, giving us figure 15. *The actual template used in the matched filtering started at 30x30 pixels, by resizing this template. Its size was changed to cover different possible scalings in our test image.*

Fig. 15. Average of 30 selected faces - Our Template/Matched Filter

B. Application of Template for Face Detection

Our basic algorithm at this stage may be summarized as follows:

1) Resize the image through appropriate filtering and subsampling so that the smallest head in the resulting image is likely to be no smaller than the initial size of our template, 30x30 pixels
2) Convolve the masked grayscale image (figure 12) with the template. Normalize the output by the energy in the template.
3) Look for peaks in the resulting output and compare them to a given range of thresholds.
4) Any pixels that fall within the threshold range are deemed to be faces and are marked as such. In order to help prevent the occurrence of false detections and multiple detections of the same face, we subsequently mask out the pixels in the reference grayscale image (figure 12) with a small vertical rectangle of a size comparable to the template and large enough to cover most of the detected head and neck regions.
5) The threshold range is reduced to a preset lower limit. Apply another stage of convolving. If the lower limit is already reached, proceed to the next step below.
6) In order to detect larger scale faces, the template is enlarged and the thresholds are reset to the upper limit. We again go through the convolution, detection, threshold reduction, steps.
7) If an upper scale limit is reached, quit.

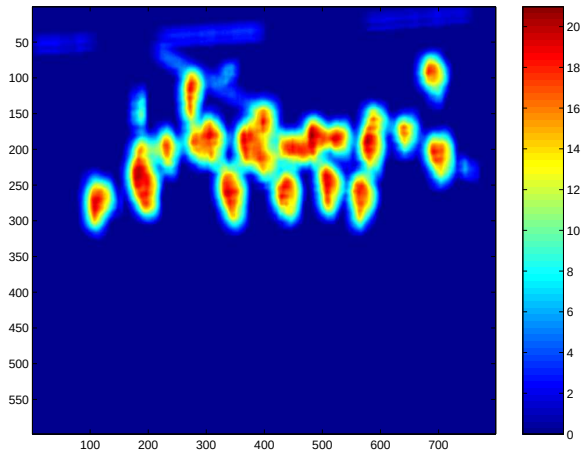

Fig. 16. Example of a typical application of the template to the masked image (2D view)

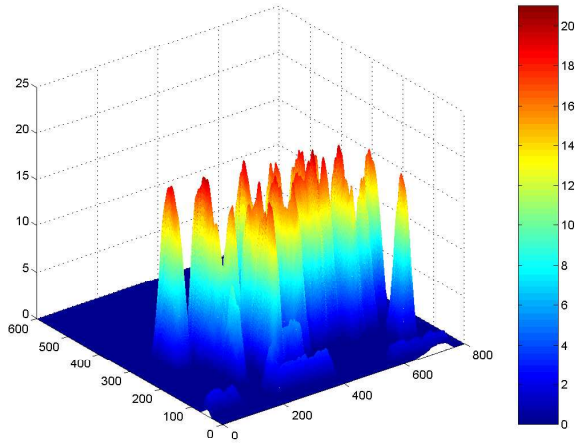

Fig. 17. Example of a typical application of the template to the masked image (3D view)

Fig. 18. Example of the masked image after detection of possible faces. Surrounding pixels of high likely association with the detected pixel(s) are masked out to avoid future multiple detections and false alarms.

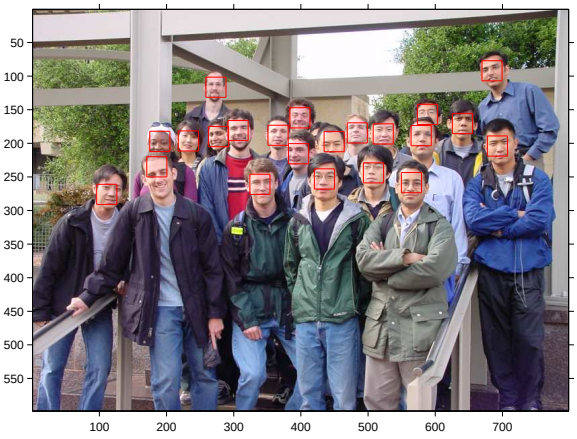

Fig. 19. Final result of face detection algorithm

For an example of what the results are of a typical application of the template matching step, please refer to figures 16 and 17. We see that there are peaks at the locations of the faces, but that due to the proximity of the faces that the peaks are closely located in space.

As mentioned in the steps of our algorithm, when a given peak was determined to be a face pixel by virtue of falling into the threshold interval, we then remove any pixels that have a high likelihood of being associated with that single face from the masked image. An example of what masked image results from this may be seen in figure 18.

It is worth mention that we found that we were able to detect tilted faces with reasonable reliability without necessarily having to rotate the template through a range of different angles. It is worth asking the question of what sorts of results we would have received had we used a template with a shape similar to that of a face...perhaps a Gaussian shaped template. It leads to the question of how much of our results are due to ”face detection” as opposed to just ”pixel energy detection”. These are questions that we hadn’t sufficient time to investigate, although the question is a relevant one in interpreting our results. Without addition experimental data, we will have to refrain from further comment on this point.

VI. Results and Conclusion

Using the algorithm described in the previous section has produced rather reasonable results when applied to the various training images. For the particular image that this report has been looking at, we were able to accurately detect the 22 present faces and had no false alarms or misses. The results may be viewed in figure 19.

Our results when applied to the other testing images ranged from approximately 85% to 100%. We are looking forward to seeing how it performs when applied to future test images. Our algorithm is reasonably fast in that it performs typically in approximately 100 seconds or so and is sufficiently accurate given the difficulty of the problem.

We will note that the distribution of work was even between both team members, with both members contributing to all aspects of the project it fairly equal amounts.

VII. Acknowledgements

We would like to thank Dr. Bernd Girod an Chuo-Ling Chang for excellent instruction throughout the quarter. We

both enjoyed the material considerably and and are looking forward to taking related courses in the future. Thank you very much.

VIII. Bibliography

References

[1] R. Gonzalez and R. Woods, *Digital Image Processing - Second Edition*, Prentice Hall, 2002.
[2] B. Girod, *Lecture Notes for EE368*, Spring 2002.
[3] M. Elad et al., *Rejection based classifier for face detection*, Pattern Recognition Letters, 23, 2002.