Face Detection

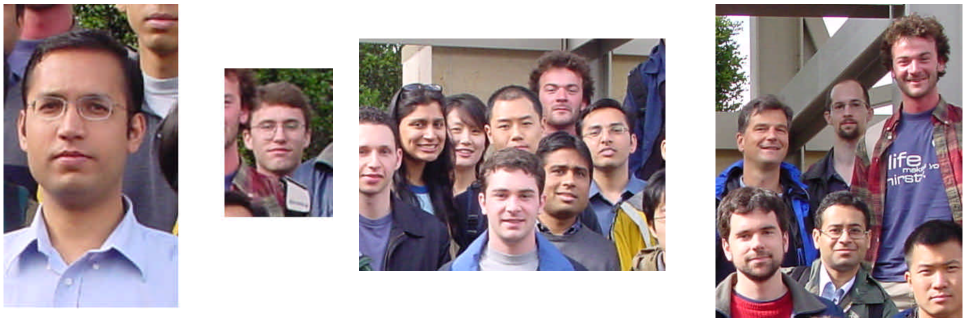
EE368 Final Project
Spring 2003

Peter Brende
David Black-Schaffer
Veniamin Bourakov

Introduction

The final project for EE368 this year consisted of the challenge of correctly locating the faces of the students in the class in several outdoor photographs, with extra credit being awarded for identifying females. The images were provided at quite high resolution and were taken with reasonably consistent lighting but the relative sizes of the faces varied with the degree of zoom of the camera. Most of the images further exhibited significant overlap of faces with little or no border between them.

Figure 1 --Typical image characteristics



Scale differences **Face overlaps** **Lighting and color variations**

The algorithm designed to detect these faces was ultimately based on a face-shape template match on color-separated skin regions, and performed admirably. By adding more templates for the shape matching, we were able to fine-tune the performance as measured on the training images. It should be noted that the algorithm is designed specifically for this particular image set as it depends on the segmentation of skin regions based on the observed skin-color distributions in the training images.

Algorithm Design

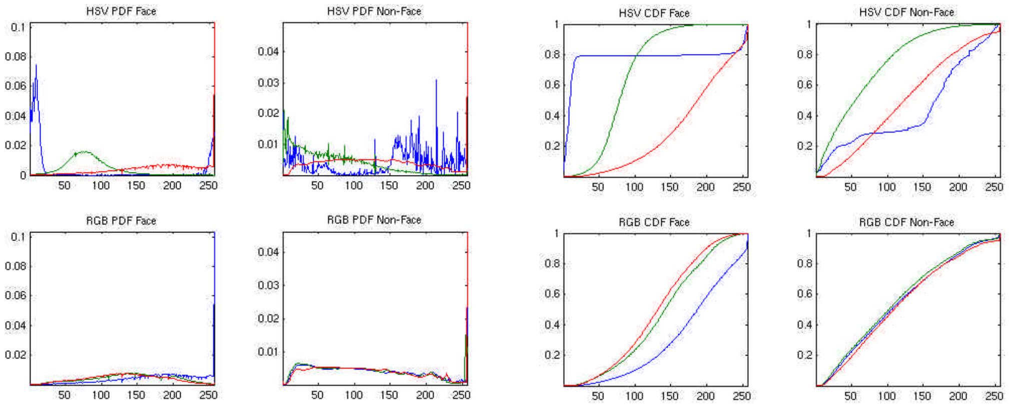
Before designing our algorithm we read through the reports of the groups whose algorithms functioned well last year. We determined that all of these algorithms operated by first doing a skin/not-skin separation based on color, then a series of morphological operations to separate face blobs in the skin/no-skin image, and finally some sort of detection based on the image represented by each blob to determine if it was indeed a face. We also came away with the understanding that the exact template used for the matching was much less important than the overall shape and size. Based partly on the results from last year, but also on our initial experimentation, we concluded our algorithm should begin with color segmentation and then be followed by spatial feature analysis involving either morphological processing or template matching.

Color-space Separation

The most salient face/non-face information in the images comes from the color. The areas of skin in the images clearly demarcate the regions where one can expect to find faces. To find these regions, color statistics were first collected on all the face and non-face pixels using the provided reference masks. This allowed for the calculation of the three marginal probability distributions corresponding to the three components of the color-space. In general, estimating an arbitrary probability distribution is difficult and requires assumptions of a parametric form such as binomial, uniform, or Poisson. In our case, however, we have a very large amount of data, which allows us to use a brute force approach and estimate the probability of an outcome to be the observed frequency of that outcome. In other words, a histogram of the observed color values serves very well as an estimation of the true probability distribution [2].

Examining these histograms (see Figure 2) revealed that in the HSV color-space the skin color lay almost exclusively in the region very close to H=0, i.e., the red region. (Hue is the blue line in the top graphs.) As the hue value wraps around, this effect can be seen by the large blue peaks on both the left and right, and by the large jumps near 0 and 255 in the CDF for the hue in Figure 2. However, a simple selection based purely on this criterion picked up much of the bright parts of the background due to the non-linear nature of the HSV space. That is, these bright background areas do not have a well defined hue, so their H value was mapped to zero, thus bringing them into this definition of skin color under a simple skin classifier. A slightly more sophisticated mapping that also looks at the saturation was able to obtain reasonably accurate skin masks with little effort. Unsurprisingly, examination of the RGB space did not provide any obvious clues as to how to separate out skin color.

Figure 2 -- PDFs and CDFs for HSV and RGB Color-spaces

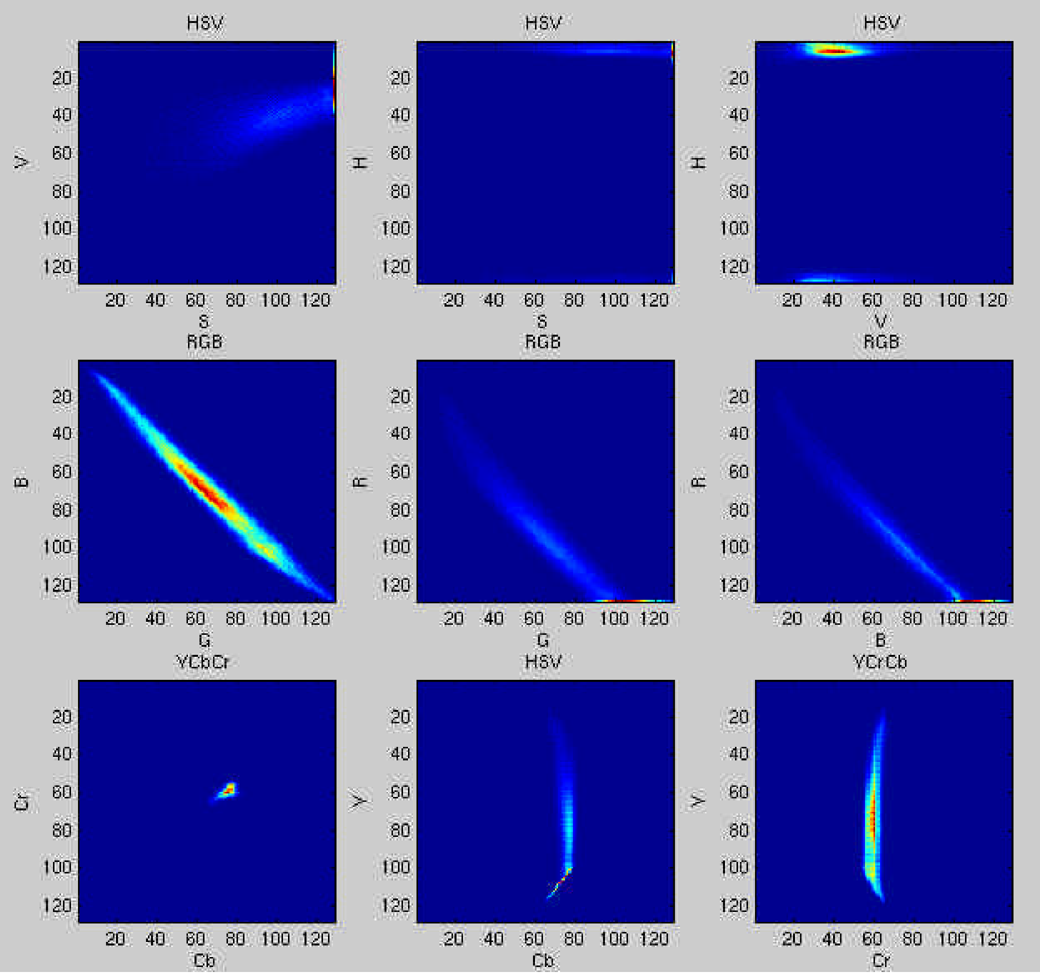


Face vs. Non-Face Color PDFs for RGB and HSV Color-spaces

Face vs. Non-Face Color CDFs for RGB and HSV Color-spaces

While this approach provided reasonable (and very fast) skin masking, it picked up on many elements in the image that were less desirable. A more sophisticated method was developed which applied Bayes rule to the full 3-dimensional joint probability distributions of color values for face and non-face regions. As this method did not rely on the separabiltity of the face/non-face colors in color space, it was implemented in RGB color-space to save the time of doing an HSV transformation. Since there is essentially a one-to-one mapping between the two color modes, results obtained from using Bayes rule on the full joint probability distribution will not be affected by choice of color space. It should be noted that because the face/non-face data was collected using the reference image masks, any hands or arms present in the images were counted as being in the non-face category despite the fact that they were virtually identical to the face colors. However, given that the vast majority of the skin-colored pixels were in the face regions this did not have much effect on the results.

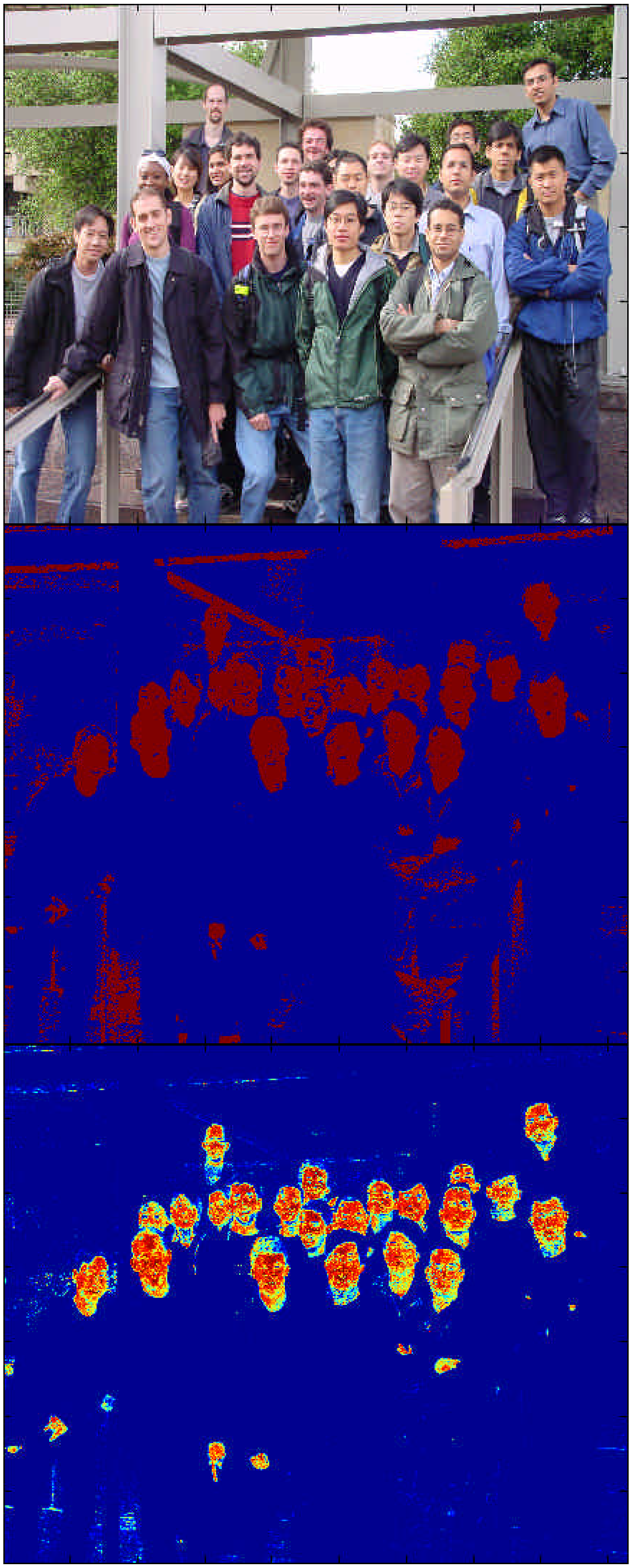
Figure 3 -- Slices of the three-dimensional complete face PDF for HSV, RGB, and YCrCb color-spaces



Note: It is clear from this figure why a simple linear separator can be so effective in the HSV space as all the skin pixels are clustered very tightly in the hue component as can be seen in the upper right and upper middle graph.

The final color-space separation ultimately used was one which did a table lookup via the MATLAB interp2 function for the conditional probability of a pixel being a face pixel given its color. The matrix for this PDF was 128x128x128, which took up 17MB of memory. This resulted in a very high quality mask, which accurately selected out skin-colored elements, a few bits of jacket, and a small section of the tile roof in the back of image 3, with a very reasonable computation time of less than ten seconds. (See Figure 4.)

Figure 4 -- Comparison of Skin masking techniques



Original Image

Initial Mask

Using only simple HSV criteria this mask catches much of the support beams in the background and the gravel on the bottom.

Conditional Probability Mask

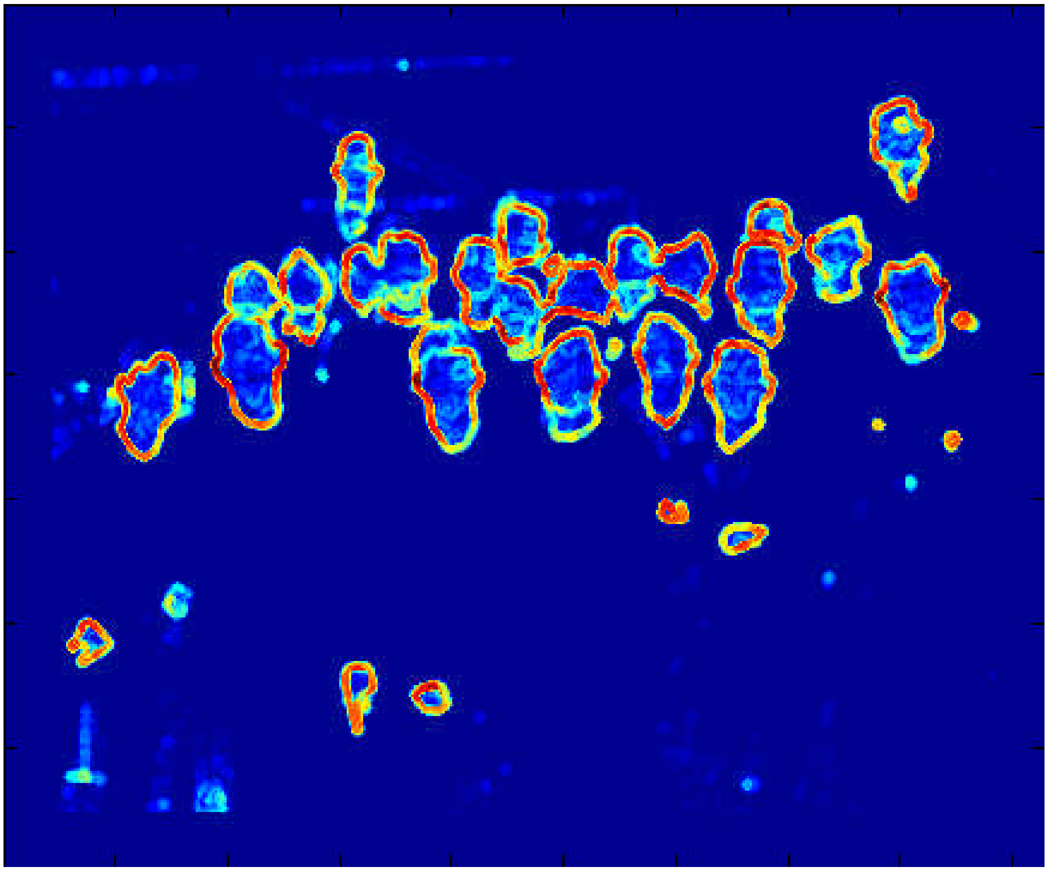
This mask was generated by a table lookup of the conditional probability of a given pixel being a face given its color. The lookup was implemented very quickly using the MATLAB interp2 function and the results are generally excellent.

Morphological Object Analysis

Once we have mapped the full color image to the ‘skin-probability’ image, the next step is to identify the different blob-like regions as faces. The first approach we consider involves morphological processing of a thresholded version of the skin-probability image. The aim or morphological object detection is to convert the binary image of the face regions (with each face possibly broken into many sub-regions), into an image where each distinct blob represents exactly one face. This method proved to be significantly more difficult than students had found it last year as many more of the faces overlapped one another. To successfully separate them required a degree of erosion, which eliminated many of the smaller faces. This led to the consideration of doing multiple passes to try and detect smaller and smaller faces, but, alas, we could not solve the problem of determining whether a given blob was part of a larger region representing a face or was a face in itself. Eventually progress in other areas of the algorithm convinced us that accurate blob separation was not crucial.

A different approach was tried wherein dilation and erosion operations were performed on the face probability image to produce edges around the faces. Then a correlation was performed with several handpicked prototypical edge templates, and points that exceeded a given threshold were used to determine face hits. When a face was found, the corresponding part of the original skin-probability image was zeroed out to prevent further double-hits. While this approach worked reasonably well, it suffered from the requirement of many distinct outline templates to match the wide variety of edge shapes (see Figure 5). Ultimately, our use of morphological techniques was abandoned for the template matching discussed below.

Figure 5 -- Edge-map from the face-probability image by erosion and dilation



Template Matching

In the end, the essence of our face detection algorithm was rooted in template matching. The important issues to consider in a template matching routine include the type of image to use (luminance, skin-probability, binary mask etc.), the number and nature of the templates to be used, and the threshold values. The different tradeoffs will be discussed below.

Basic Theory of Template Matching

Template matching is a process of locating in a visual scene *s(x,y)*, an object that is very similar to one represented by the template image *t(x,y)*. We use the *mismatch energy* criteria function to determine the quality of the match within the scene [1].

$$E(p,q) = \sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}[s(x,y) - t(x-p,y-q)]^2$$

$$= \sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}|s(x,y)|^2 + \sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}|t(x,y)|^2 - 2\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}s(x,y)\cdot t(x-p,y-q)$$

To achieve a minimum, it is sufficient to maximize area correlation, which is equivalent to convolution of the image *s(x,y)* with the template *t(-x,-y)*

$$r_{st}(p,q) = \sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}s(x,y)\cdot t(x-p,y-q) = s(p,q) * t(-p,-q)$$

From the Cauchy-Schwarz inequality it follows that the cross-correlation *r(p,q)* attains the maximum value when the position of the template coincides with the image. Therefore, the object *t(x,y)* can be located in the image by searching for the peaks in cross-correlation function. In general, object finding via template matching requires a large variety of templates, since the size, angle, and rotation of the search object is not known exactly. Thus, computational speed can become a limiting factor. Consequently, it is preferable to exploit the properties of the Fourier transform and compute the convolution through multiplication in the frequency domain.

$$R_{st}(w_x,w_y) \equiv F\{r_{st}(p,q)\} = S(w_x,w_y)\cdot T^*(w_x,w_y)$$

Implementation of Luminance Image Template Matching

Initially, our intuition led us to believe that it would be possible to use template matching with the grayscale luminance image to isolate the locations of the different faces after some preprocessing based on color segmentation.

As no template matching can be performed without a template, the first step is to obtain one. Facing the unpleasant task of manually creating a template from a given image set, which is more time-consuming than justified by our needs, we decided to simply recruit one from the Internet. A face template with histogram equalization is shown in Figure 6a. It was found that better results were obtained when the face template was cropped to contain the significant facial features while eliminating as much of the background as possible (see Figure 6b).

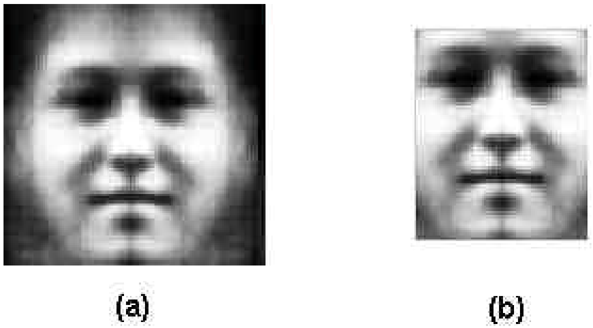


Figure 6 -- Luminance image face templates

Prior to the template correlation with the luminance image we performed an "AND" operation of the original image with the mask obtained from color segmentation in order to remove non-face regions. The template was used at multiple scales and rotation angles. For each template correlation, the points that exceeded a certain threshold were recorded for observation and analysis.

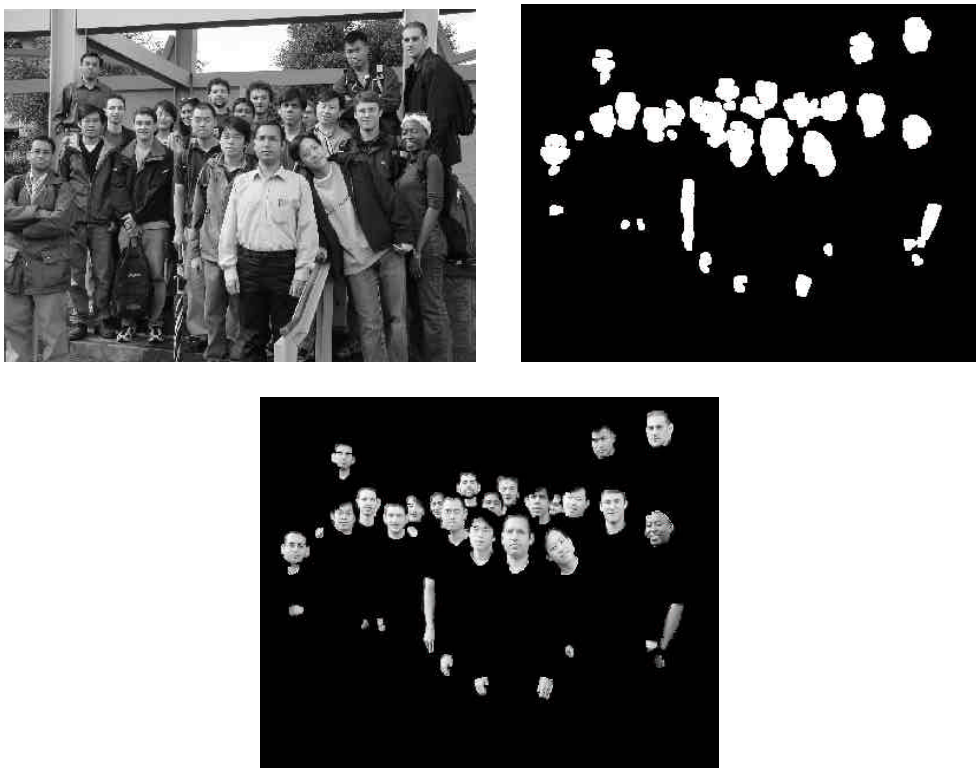


Figure 7 -- Original image, color segmentation mask and the resulting masked image

By observing the set of recorded points, we found that this method produced a number of false detections and some missed faces. A second approach described below, being developed by our group in parallel, had begun to show great promise. Thus, we abandoned work with the luminance image for template matching with the skin-probability image.

Template Matching of Skin-Probability Image

The basic difficulty with template matching in the luminance image is that the correlation between two visual objects with fine features is generally very sensitive to non-linear operators such as rotation, scaling and change in projection, all of which are active distortion mechanisms in our images. Say, for example, that a template is extracted directly from one image and used to detect faces. The detector will of course find that same face, but it is much less likely to find others – even the face of the same person in a different image. There are several possible approaches to this problem: 1) Use a 'database' consisting of a very large number of templates with the hope that for any face in the image, there will be a similar face template in the database. 2) Use only a few general templates that capture the rough characteristics of the face, such as the basic oval shape, and the darkness in the eye and mouth region. We felt that the second of these approaches was more reasonable, especially in light of the computational time constraints.

At this point, we realized that by using a rough, blurred template, we have essentially reduced our method to searching for oval shapes of skin. We reasoned, then, that it would be advantageous to use an image that contains only information about the presence or absence of skin in a given region. Thus, we decided to try performing the template matching directly on the skin-probability image. The primary advantage of using the skin-probability image is that is greatly simplifies the information content; the skin-probability image emphasizes the information we are most concerned with – the presence or absence of skin, while de-emphasizing other information such as variation in saturation or brightness. A secondary advantage of simplifying the information content is that it becomes easier to both design and debug the algorithm. Finally, by working directly with the skin-probability image, we are able to sidestep the issue of choosing a threshold for creating a binary skin/not-skin mask (see the upper-right image in Figure 7). A binary mask has the undesirable characteristic of 'infinitely' amplifying skin classification errors.

Of course, by mapping the image to 'skin-probability space' we lose some information, such as the location of specific facial features like the nose and mouth, as well as information about the boundary between two overlapping faces. In the end, though, we found the value of the simplified approach to outweigh the loss in detail.

General Strategy

Our basic assumption under the skin-probability template-matching algorithm is that face regions and only face regions have a strong value in the skin-probability image. (This assumption neglects the existence of hands and arms; however, we will use a simple preprocessing routine to eliminate these skin regions from consideration.) Therefore, our task becomes one of assigning each pixel of the dominant skin regions (see Figure 4) to a unique face. In each iteration of our algorithm, we attempt to match a face template to a face region of the image. If a face is found, we eliminate that region from further consideration by 'subtracting' the template from the image. The process continues until all significant skin regions have been assigned to a face.

One can imagine how a mistake made early in the iterative process can lead directly to errors in the later rounds. On one hand, if too small of a face is detected, the subtraction step may leave a portion of the face, which might then be falsely recognized as another face later on. On the other hand, if too large of a face is detected, the subtraction step may actually eliminate two heads from the skin-probability image. In other words, the key to our algorithm is to use many different scalings of templates so as to ensure that the template and the matching face-skin region are of approximately the same size.

To avoid the first problem discussed above, we *cannot* begin with small templates, since there are many small blobs/ovals within a larger blob, which would register as hits. Therefore, in our algorithm, we begin with the largest scaling of our templates and decrease the size at each iteration. The solution to the second problem mentioned above is to set the detection threshold for the correlation value as high as possible, without sacrificing overall sensitivity. This prevents a large template from being matched with a smaller face.

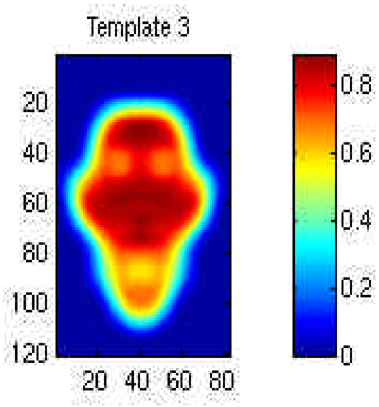


Figure 8 -- Basic template shape with neck

As a final consideration, it is desirable to have a close match between the shape of the template and the corresponding face-skin region. We found that the greatest variation in shape occurred when the skin of an individual's neck was visible. With this in mind, we constructed two templates, one in the basic face shape, and a second one with the neck region present as shown in Figure 1. In the figures below, we get a sense of the steps that take place within each iteration of the template matching.

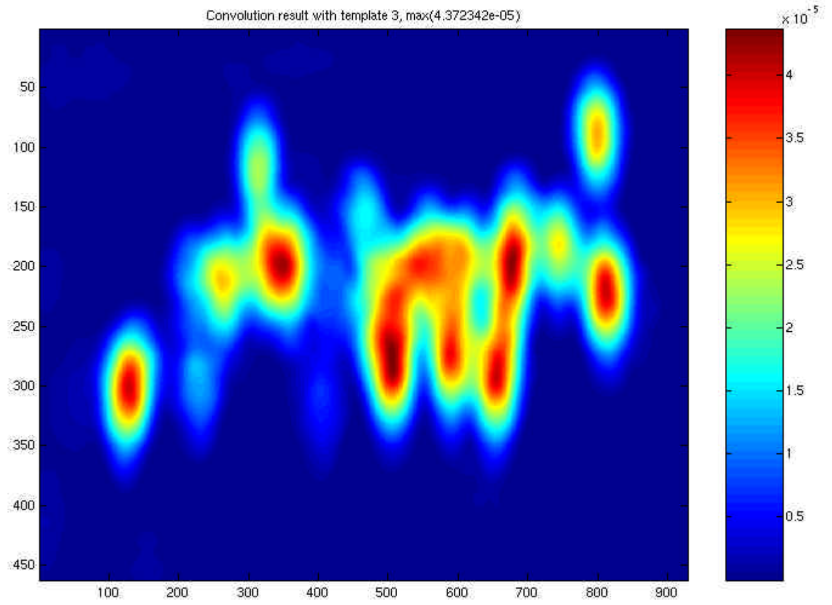


Figure 9 -- A typical correlation result between template and skin-probability image

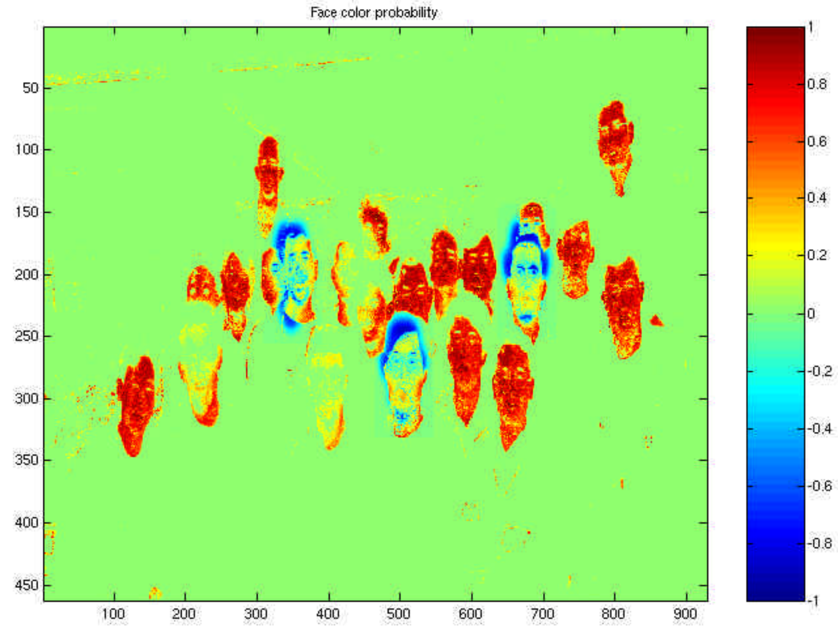


Figure 10 – Skin-probability image after 'subtraction' of 3 detected faces. Note that several other faces in the image have already been 'subtracted' in previous iterations of the algorithm.

Choosing Correlation Thresholds

When using a large number of templates, as in our case, choosing the correlation threshold value to be used with a given template appears a daunting task. Fortunately, in our case, all of the templates were obtained by resizing the two basic templates, which allowed us to simplify the training process.

Consider two spatially continuous images $s_{cont}(x,y)$ and $t_{cont}(x,y)$ with cross-correlation function $R_{st}(p,q)$. If we spatially-scale the two images by a factor a, the new cross-correlation function is related to the old by $R_{scaled} = a^2 * R_{st}(p/a,q/a)$. This indicates that the peaks of an auto correlation function are scaled by the square of the scaling factor *a*.

Now, if we make the assumption that the different face regions in our image differ only in size, then it would follow that the correlation threshold value should be adjusted with the square of the scaling factor *a*. In reality, the smaller face blobs in the skin-probability image tend to be small because these faces are obscured by other faces. Since the image of an object is altered when it is partially obscured, our assumption about constancy in the shape of the face regions is violated. We deal with this reality by hand-tuning, through trial and error, the correlation threshold values around a 'predicted' value given by $a^2 ?_{thres}$. The parameter $?_{thres}$ is chosen so that a correlation of the 'basic' template with a face-blob of similar size in the skin-probability image produces a peak that just exceeds $?_{thres}$.

Algorithm Implementation

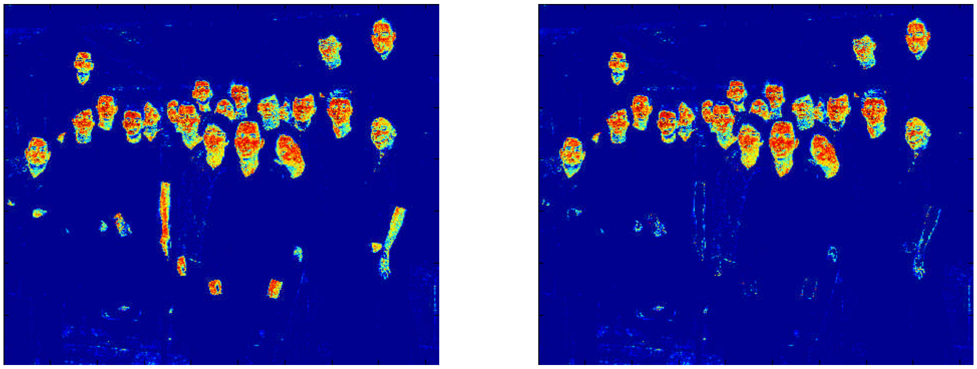
1. Load probability and template data
2. Down-sample the image by factor 2:1
3. Calculate the face-probability image by color
4. Remove hands/arms
5. Template match with skin-probability image
6. Eliminate false positive hits on necks of large faces
7. Remove detections at the edge of the image
8. Remove patterned hits

Step 1. Load probability and template data. The conditional probability matrix for the skin/not-skin color-space separation was pre-computed and stored in file, which is then loaded at the beginning of each classification execution. This data is used to calculate the skin- probability image. The template data consisting of 13 scaled versions of the two basic templates along with their associated thresholds are also loaded at this time.

Step 2. Down-sample the image by 2:1. As the image is extremely high resolution the first thing we do to it is to down-sample it to half the horizontal and vertical size by simply throwing out every other sample. We do not use an anti-aliasing filter, as we are not concerned with the fine detail of the image; the resolution is sufficiently high that we do not see any degradation in our outputs from this approach. Yet, by decreasing the number of pixels by a factor of four, the computational time for subsequent operations is decreased greatly.

Step 3. Calculate the face-probability image by color. This technique was described earlier in this report and simply uses MATLAB's interp2 function to do a table lookup on the pre-computed and pre-loaded conditional probability of a pixel being a face pixel given its color.

Step 4. Remove hands/arms. The skin-probability image is thresholded and smoothed with a median filter to generate a binary image of skin blobs. Each of the blobs below a certain threshold in the vertical dimension is examined and removed if it is either too small (a hand) or has too high of an eccentricity value (an arm). The thresholds were chosen based on typical values seen in the training images. The effectiveness of this step is seen below in Figure 11.



Face-Probability Image

Hands and arms removed below a given threshold.

Figure 11 -- Skin-probability image after hand and arm removal

Step 5. Blob detection through template matching. The principles of this technique were described earlier in detail. The specific implementation here has 13 templates, of which the first 9 are scaled versions of a symmetric head-with-neck image and the last 4 are scaled versions of a symmetric head-without-neck image. The thresholds were manually selected for the best results. The actual implementation simply consists of taking the correlation with the original image of a given template, finding the peaks above that template's threshold, marking those as faces, and subtracting the template from the remaining face-probability image to prevent future detections of that particular face.

Step 6. Remove no-neck heads below neck-heads. In general the faces in the bottom row (the ones closest to the camera) are larger than those in the back. If these faces are rotated they do not match our template and the larger neck templates leave a large piece of the original rotated image behind. This is often detected incorrectly in subsequent iterations as a face by the smaller no-neck templates. As all the faces towards the bottom of the image are assumed to be closer to the camera we can be quite certain that there should be no small, obscured faces in this region, and we remove any false hits of this type.

Step 7. Remove edge points. This step simply removes any points within 4% of the edge of the image. This is appropriate for our images (and removes some of the roof hits) but is inappropriate for the images from last year, which have many faces on the sides.

Step 8. Remove patterned hits. This step is necessary to remove the tile area on the roof seen in several of the images. This area is both large and has a very face-like color, resulting in several strong hits with many of the medium-sized templates. We felt that the strong repeating pattern of the roof tile might be evident in the Fourier transform of a sub-image taken from that region. Indeed, the results of a 2D FFT display considerable spikes at the frequency corresponding to the repeating pattern.

By zeroing the low frequency components of the FFT and then searching for peaks exceeding a certain threshold, we are able to eliminate false positive detections in the roof region.

Results

Table 1 -- 2003 Image Results

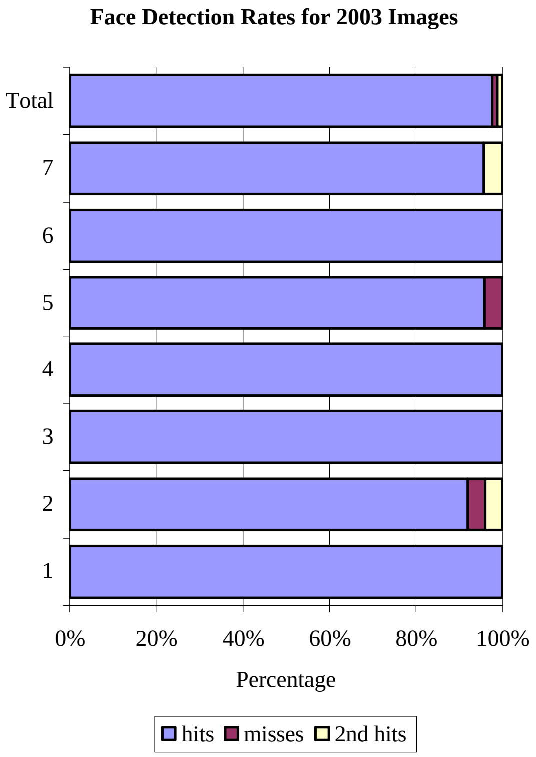
Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Total**
hits	21	23	25	24	23	23	22	**161**
misses	0	1	0	0	1	0	0	**2**
2nd hits	0	1	0	0	0	0	1	**2**
faces	21	24	25	24	24	24	22	**164**
Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Total**
%hits	100%	96%	100%	100%	96%	96%	100%	**98%**
%misses	0%	4%	0%	0%	4%	0%	0%	**1%**
%2nd hits	0%	4%	0%	0%	0%	0%	5%	**1%**

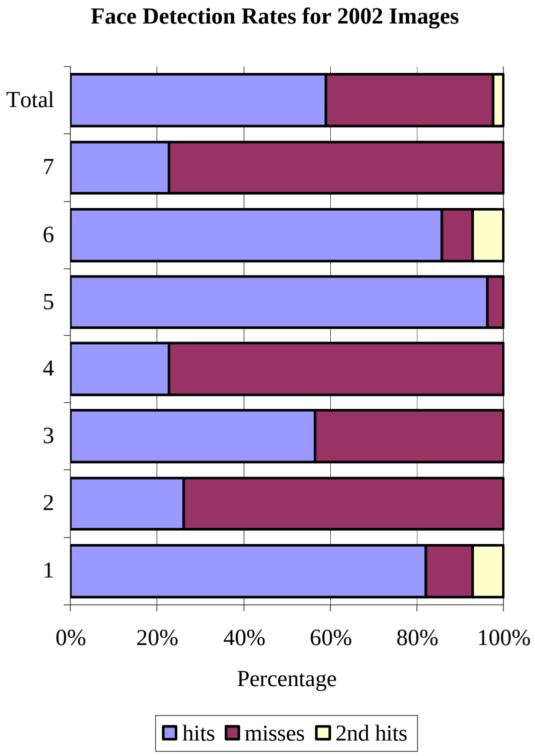
Performance on this year's images was excellent. With the exception of two missed faces and two faces that received double hits the detection was perfect for an overall rate of 98%.

For fun we tried our face detector on last year's training set with mixed results. For the images with roughly the same size faces in similar lighting the results were excellent, but for the images with much smaller faces, or which had significantly different face colors, the detector was not sensitive enough. Further, by throwing out points too close to the edge we eliminated several of the faces cut off at the edge of last year's images. Changing our algorithm to adjust to the intensity of the detected faces and adding more small templates in that case would have allowed us to be robust enough to do better on these images.

Table 2 -- 2002 Image Results

Image	1	2	3	4	5	6	7	**Total**
hits	23	6	13	5	26	24	5	**102**
misses	3	17	10	17	1	2	17	**67**
2nd hits	2	0	0	0	0	2	0	**4**
faces	25	23	23	22	27	26	23	**169**
Image	1	2	3	4	5	6	7	**Total**
%hits	92%	26%	57%	23%	96%	92%	22%	**60%**
%misses	12%	74%	43%	77%	4%	8%	74%	**40%**
%2ndhits	8%	0%	0%	0%	0%	8%	0%	**2%**





Conclusions

As the initial step of the algorithm, color-space separation was by far the most effective means of eliminating non-face regions from consideration. For the subsequent face-segmentation step, we found that the very simple method of looking for face-like shapes within the skin-probability image to be effective and computationally efficient. We did not have much success with morphological processing nor detection based on actual face features such as they eyes and mouth. These more sophisticated approaches certainly have merit in a more general-purpose face detection program. However, with our very consistent and predictable set of training images, the simplest approach proves to be more than adequate, as is evidenced by the overall accuracy rating of 98%, and quick execution time of roughly one minute.

Bibliography

[1] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Addison-Wesley, 1989.

[2] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. John Wiley & Sons, Inc, New York, 2001.