Detecting Human Faces in Color Images

J. Cai & A. Goshtasby
CSE Department
Wright State University
Dayton, OH 45435

C. Yu
EECS Department
University of Illinois
Chicago, IL 60607

Abstract

A method is introduced that detects human faces in color images by first separating skin regions from nonskin regions and then locating faces within skin regions. A chroma chart is prepared via a training process that shows likelihoods of different colors representing the skin. Using the chroma chart, a color image is transformed into a gray scale image, with the gray value at a pixel showing the likelihood of the pixel representing the skin. By segmenting the gray scale image, skin regions are separated from nonskin regions. Then, using the luminance component of the color image and by template matching, faces are located within skin regions.

1. Introduction and Background

Indexing and retrieval of video images containing human activities require automatic detection of the humans and, in particular, human faces in images. Techniques that recognize faces or analyze facial expressions also require knowledge about the existence of faces in images. In this paper a new method for detecting faces in color images is presented.

Majority of existing face detection techniques rely only on image gray values [2] [3] [8] [12] [14] [17] [18] [19] [20] in spite of the fact that most images generated today are color. A small number of techniques use color information to detect faces in images [1] [5] [13] [16].

Methods that are based on image gray values detect predefined image features and use them either in a system that has a learning ability or in a model-matching algorithm to detect the faces. A method developed by Rowley *et al.* [14] tests for existence of faces of different sizes at each image location using a neural network. Although the system is designed for detecting frontal faces in images, the network filters can be trained to detect side-view faces also. An alternative to neural networks is a probabilistic visual learning algorithm designed by Colmenarez and Huang [2]. In a system developed by Sung and Poggio [17], first a model face is generated using an example-based learning method. Then at each image location, a feature vector is computed between the local image and the model. Finally a trained classifier is used to determine whether a face exists at that location or not.

Many methods rely on feature matching to detect faces. A method developed by Yow and Cipolla [20] uses second derivative Gaussian filters, elongated at an aspect ratio of 3:1, to locate horizontal bar-like features in images. From the bar-like features, elongated facial features such as the eyes and the mouth are located, and, by grouping the features and matching their relations with those of a facial model, instances of faces are found. By changing the size of the Gaussian filter, this method can be used to detect different-sized faces in images. A similar method is described by Cootes and Taylor [3], which instead of bar-like features uses statistical features, and by matching relations between detected features and those of a model face, finds faces in images. A method developed by Leung *et al.* [12] uses a graph-matching method to find likely faces from among detected facial features. First, graphs representing candidate faces are generated from detected facial features, and then by a random graph-matching algorithm, true faces are located from among candidate faces.

A method that considers a facial pattern a texture pattern and uses texture features to locate faces in an image is described by Dai *et al.* [6]. Texture features based on a space gray level dependence (SGLD) matrix [9] are used. By comparing texture features of a model face with texture features of windows in the image, potential faces are located. Using model faces and windows of different sizes, this method can detect different-sized faces in an image. A method that uses only image edges to detect faces is described by Govindaraju [8]. In this method, second-derivative edges are first determined. Then using models of human face

curves, faces are located within the edges.

A small number of techniques use color information to detect faces. These techniques first select likely image regions for faces. Then they detect faces in the selected regions using facial patterns. Dai and Nakano [5] isolated and kept orange-like regions in the YIQ color space as the skin regions and eliminated the remaining regions. Then they used texture features in gray scale images [6] to identify faces in the skin regions. Chen *et al.* [1] prepared a color chart in the HSV color space that represented possible skin colors. Then they used the color chart to identify image regions that represented the skin. Using three face templates, they located faces in skin regions through a fuzzy pattern-matching algorithm. Face templates of different sizes were used to detect different-sized faces in an image. Sobottka and Pitas [16] detected skin regions in an image using the hue and saturation and selected regions that were elliptic as faces. Miyake *et al.* [13] used a chart of skin colors in the Luv color space to detect skin regions in an image. They then located round regions from among obtained regions to detect faces. Schiele and Waibel [15] used the red and green components of the skin color in a large number of images to construct a histogram. They then used the histogram to compute the probability that a particular combination of red and green belonged to the skin and used the probabilities to classify image pixels to skin and nonskin regions. Crowley and Berard [4] used this method to track faces in video images.

In the method proposed here, information about skin colors is used first to find likely image regions where a face may exist. Then information about facial patterns is used to determine instances of faces in such regions. The differences between the proposed and previous face-detection methods can be summarized as follows:

- Computations are performed in the 1976 CIE LAB color space. Colors in CIE LAB space are more uniformly spaced than colors in RGB or HSV spaces [11], enabling use of a fixed color distance in decision making over a wide spectrum.
- Only chroma (a and b components in CIE LAB) is used to separate skin from nonskin regions. Because of the smooth and curved shape of faces, reflected light intensity across a face varies considerably; therefore, luminance component of the color can mislead the skin detection process. Chroma across a face, however, remains relatively unchanged and can be reliably used to detect skin regions. After skin regions are detected, the luminance component of the colors is used to detect facial patterns.
- Rather than classifying pixels to skin and nonskin regions, a weight is assigned to each pixel showing the likelihood of the pixel belonging to the skin. The weights are obtained from a chroma chart that is prepared through a training process.
- Using the chroma chart, a color image is transformed into a gray scale image, with the gray value at a pixel showing the likelihood of the pixel belonging to the skin. This gray scale image is then segmented into skin and nonskin regions.
- The proposed method is based on a bottom-up approach; therefore, it can find faces of different sizes and orientations in an image. This is in contrast to top-down methods that search for faces of prespecified sizes and orientations.

The proposed face detection method consists of the following steps:

1. Determine a chroma chart using the ab components of colors in the 1976 CIE LAB color space such that the value at entry (a, b) shows the likelihood that chroma (a, b) represents the skin.
2. Transform a color image into a gray scale image with the gray value at a pixel showing the likelihood of the pixel belonging to the skin.
3. Determine skin regions in the obtained gray scale image by image segmentation.
4. Detect facial features in skin regions and hypothesize faces.
5. Verify existence of hypothesized faces by template matching.

These steps are described in more detail next.

2. Preparing the Chroma Chart

A chroma chart is used to train a vision system to distinguish skin from nonskin colors. Color charts have been used previously with considerable success [1] [13] [16]. Previous color charts, though, have been binary: colors have been grouped to either skin or nonskin. In our model, any color can represent the skin, but with a different likelihood.

Our color chart will have two components, representing the a and b values in the 1976 CIE LAB color coordinate system. Therefore, we refer to our color chart as the *chroma chart*. As mentioned earlier, we do

not use the luminance component of a color to prepare this chart because luminance may vary considerably across a person's face and cannot be a reliable measure to separate skin from nonskin regions. Chroma, on the other hand, remains relatively unchanged across a skin region and can be reliably used to separate skin from nonskin regions. Chroma has been used effectively to segment color images before [7].

Our chroma chart is obtained by mapping the ab values in the CIE LAB color space to discrete array entries. Entries of the chart are initially set to 0. To train the system, skin color samples are taken from a large number of images, the colors are transformed into CIE LAB color coordinates [11], the chromas of the colors are mapped to the array entries, and entries are changed to 1. This mapping involves transforming floating-point values for a in the range [-10.0, 40.0] to discrete array indices in the range [0, 499], and transforming floating-point values for b in the range [-10.0, 60.0] to discrete array indices in the range [0, 699]. A pair of ab values obtained from the color at an image pixel will, therefore, correspond to an entry in a 500×700 chroma chart.

To reduce the effect of noise in the samples, instead of using the color at a single pixel, we use the average color of a 3×3 window centered at the pixel. To obtain the average color, first the average red, average green, and average blue in the 3×3 window are determined. The average red, green, and blue values are transformed into Lab coordinates, and the ab components are used to identify the entry in the chroma chart. The entry of the chart is then changed to 1. We used twenty-three hundred skin samples in eighty images to construct our chart. As the number of samples increases, the chart becomes denser. Skin colors fill only a part of the chroma chart. Since adjacent entries in the chart show very similar colors, it is inconceivable that while some entries in a small neighborhood belong to the skin, other entries in the same neighborhood don't. Noting that as more skin samples are taken more entries in the chart become filled, we see the need for filling the small gaps in the chroma chart.

The small gaps in the chroma chart can be filled in a variety of ways. We, however, notice that simply filling the gaps is not sufficient. There are certain areas of the chroma chart that represent the skin with a higher likelihood than other areas as evidenced by denser samples. There are entries that are not very likely to belong to the skin because only sparse samples are available. Some of these samples could be due to noise or reflections from skin specularity. We need a chroma chart that contains likelihoods that are proportional to the local densities of the skin samples.

To obtain such a chroma chart, we center a 2-D Gaussian of height 1 at each sample point and record at each chart entry the sum of the obtained Gaussians. That is, if N skin samples $\{(a_i, b_i) : i = 1, \ldots, N\}$ are available, we let the likelihood that the chroma at chart entry (a, b) represent the skin be proportional to

$$T(a,b) = \sum_{i=1}^{N} \exp\{-[(a-a_i)^2 + (b-b_i)^2]/2\sigma^2\}. \quad (1)$$

Initially, we save $T(a, b)$ at entry (a, b). When values all chart entries are computed we then divide each chart entry by the largest value in the chart to obtain values between 0 and 1.

Function T produces higher likelihoods for areas containing a larger number of samples. Since a Gaussian changes rather slowly near its center and falls rather sharply about one standard deviation away from its center, the sum of Gaussians will fill gaps in local neighborhoods where a high density of samples exist, and will quickly approach zero in areas where no samples are available. Areas containing sparse samples will produce small values, showing small likelihoods of representing the skin.

Too small a standard deviation will produce a chroma chart whose likelihoods vary sharply between adjacent samples, while too large a standard deviation will assign high likelihoods to areas where only sparse samples or no samples are available. Parameter σ depends on the size of the chroma chart. For a chroma chart of size 500×700, we have experimentally found that σ between 11.0 and 19.0 produces best results. The chroma chart obtained with $\sigma = 13.0$ is shown in Fig. 1. Intensities in this chart show the skin likelihoods: the brighter an entry, the higher its likelihood to show the skin.

Twenty-three hundred skin samples were used to obtain Fig. 1. The samples were taken from the skin regions in eighty color images. Our samples were taken from persons of African as well as European and Asian origins. As samples from more races are included, the part of the chroma chart representing the skin enlarges. As a result, many objects with colors similar to those of the skin will be classified as the skin, and the number of regions to be processed to detect the faces increases. If the objective is to locate faces in images of persons, for example, with light-skin colors, one should create a chroma chart using samples from light-skin persons. This will not only increase the speed but also the face detection accuracy. On the other hand, if a chroma chart containing information about the skin colors of various races is used, it becomes possible to detect faces of different races and reduces the probability of missing faces. Doing so, however, will slow down the face

detection process and will increase the probability of detecting false faces.

We will use a chroma chart to transform a color image into a gray scale image such that the gray value at a pixel shows the likelihood of the pixel belonging to the skin. Then we will segment the gray scale image to discriminate skin regions from nonskin regions.

3. Detecting Skin-Color Regions

To identify image regions that correspond to the skin, we first transform a color image into a skin-likelihood image. This involves transforming the RGB values at each pixel (x, y) to Lab values, reading the likelihood from entry (a, b) in the chart, and saving it at location (x, y) in the image. When this process is repeated for all image pixels, we will obtain an image whose gray values represent the likelihoods of the pixels belonging to the skin.

Using the color image shown in Fig. 2a and the chroma chart of Fig. 1, we obtain the skin-likelihood image shown in Fig. 2b. As can be observed, regions having the color of the skin in Fig. 2a are transformed into bright regions, while regions not having the color of the skin are transformed into dark regions. The brighter a pixel, the higher the likelihood that the color at the pixel belongs to the skin. The point to note is that although the process detects regions having the color of the skin, the obtained regions may not necessarily correspond to the skin. Detected regions simply have the color of the skin. This process can reliably point out regions that do not have the color of the skin. Such regions can be safely skipped when searching for faces.

The image of Fig. 2b shows the likelihoods of pixels in Fig. 2a that have the color of the skin. Since skin regions appear as bright spots in a likelihood image, it seems that we can identify the skin regions by a thresholding process. Since persons with different skin colors could be present in an image and different skin colors have different likelihoods, it may be necessary to use more than one threshold value to detect skin regions of different persons.

We determine the threshold values for detecting skin regions in an image using the following observation. When the threshold value is high, the detected skin regions are small, and as the threshold value is decreased, the skin regions become larger. The size of a skin region may remains relatively stable under a wide range of threshold values, and at a particular threshold value, neighboring regions may merge increasing the size of a region sharply. Therefore, if we plot increase in size of a skin region as a function of decrease in threshold value, we will obtain a curve that gradually decreases up to a point and then increases sharply. The threshold value at which minimum increase in region size is observed while changing the threshold value will be taken as the optimal threshold value. The optimal threshold value, therefore, will find a region whose size is most stable under variation of the threshold value.

To implement this idea, first, local peaks in a skin-likelihood image are determined. To avoid detection of noisy peaks, the likelihood image is smoothed with a rather wide Gaussian (typically of standard deviation 8.0 or 9.0). Locally peak values are then located by shifting a 3×3 window over the smoothed image and determining window positions where the value at the center of the window is larger than those at other entries in the window. Our likelihood images are represented by floating point numbers; therefore, the possibility of obtaining same values in a window is extremely rare. In this manner, local peaks can be uniquely determined. The locations of the peaks will be used as seed points to determine the skin regions.

For each obtained peak, we threshold the likelihood image from 0.9 down to 0.1 with decrements of 0.1, and at each threshold value determine the size of the region containing the peak. As the threshold value is decremented, changes in region size are recorded, and the two consecutive threshold values producing the least change in region size are used to determine the optimal threshold value. If threshold values t and $t - 0.1$ produce the least change in size of a region, the optimal threshold value for segmenting the region will be $t - 0.05$. The optimal skin region in Fig. 2b was obtained with threshold value 0.25. The result shown in Fig. 2c is for one of the regions. The process can be repeated to detect other regions in the image. It is possible that several peaks fall in the same region, thus producing one region.

To summarize, the steps for segmenting a skin-likelihood image are:

1. Find local peaks in the skin-likelihood image.

2. For each peak, start from threshold value 0.9, and decrement the threshold value with steps of 0.1 until threshold value 0.1 is reached. At each threshold value, determine the region containing the peak and find the difference of its size and the region size obtained at the previous threshold value. If current threshold value is 0.9, there is no previous threshold value; so let the region size at the previous threshold value be zero.

3. Find two consecutive threshold values t and $t-0.1$ in the range 0.9 and 0.1 where change in region

size is minimum. Then, set optimal threshold values to $t - 0.05$.

Some detected regions may be very small and unlikely to represent faces. Such regions are removed before searching for faces. These regions often correspond to different parts of the scene that have the color of the skin.

Not all detected regions contain faces. Some correspond to the hands and other exposed parts of the body, while some correspond to objects with colors similar to those of the skin. Next, we will show how facial features are used to locate faces in skin regions.

4. Detecting Faces in Skin Regions

The main objective in identifying skin regions in an image is to reduce the search space for faces. Naturally, faces are in skin regions and there is no point looking for them in nonskin regions. Since facial features that do not have the color of the skin, such as the eyes, appear as small regions within skin regions, these features can be used to generate hypotheses about the positions of faces in skin regions, and their existence can be verified by template matching.

Since features such as the eyes or the mouth are elongated, we will compute their axes of minimum inertia (their major axes) [10] and use them to hypothesize the orientations of the mouth and the eyes. A skin region may contain zero, one, two, three, or more than three regions. We will call a region within a skin region a *feature*.

If no feature is detected in a skin region, we determine the width of the region and match a face model of the same width to it. To determine the width of a region, we scan along the major axis of the region and in the direction normal to it, determine the distance between the two borders of the region and take the longest such width as the width of the region. We also let the point on the major axis where maximum region width is obtained represent the center of the region. Estimating region width and region center in this manner is motivated from the observation that in a frontal-view image, the distance between the ears is usually the longest skin cover in a face region, and in a side-view image, the distance between the ear and the tip of the nose is usually the longest horizontal skin cover in a face region. The center of our frontal face model is obtained by connecting the ear centers and finding the point midway between the two. The center of a side-view model is obtained by connecting the center of the ear to the tip of the nose and finding the midpoint between the two. When no facial features are obtained in a region, the matching process involves scaling the model faces to the same width as the width of the region and placing the center of the scaled model on the center of the region and determining the match rating between the image and the model. If the match rating is above a prespecified threshold value, we classify the region as a face.

The frontal-view model to be used to verify the existence of a face in an image is shown in Fig. 3b. This model was obtained by averaging intensities of sixteen frontal view faces of men and women that had no facial hair or glasses. We used images from the Yale University and University of Bern face databases to obtain this model. The face images were geometrically transformed so that the centers of the eyes and the mouths in the images overlaid. Then the images were averaged. The averaged image is shown in Fig. 3a. Next, the interior portion of the face, as shown in Fig. 3b, was cut out. The left and right image borders in Fig. 3b were placed at the centers of left and right ears. We also prepared a side-view model of the face by geometrically transforming sixteen images from the Yale University and University of Bern face databases in such a way that the centers of the eyes, the mouths, and the ears in the images overlaid. We then averaged their intensities, and the portion of the face that was common to all images was cut out as a side-view model. The left and right image borders corresponded to the tip of the nose and center of the ear. This model was reflected about the vertical axis to obtain the model from the opposite view. The side-view models are not shown here. The model of Fig. 3b will be used to verify existence of frontal-view faces, while the two side-view models will be used to verify the existence of side-view faces in images. The reason for cutting out the central portion of image 3a is to exclude portions of a scene that do not belong to the face or the portion of the face that varies from one individual to another.

Since our model is a gray-scale image, we will only use the luminance component of a color image in template matching. After overlaying a model on a gray-scale image, we compute the cross-correlation coefficient between the pixels in the model and the image and use it as the match rating between the model and the image. When computing the cross-correlation coefficient, only nonzero pixels in the face templates of Fig. 3b are used. This is to reduce the effect of scene contents other than the face on match rating. Since the cross-correlation coefficients vary between -1.0 and 1.0, our match ratings will vary between -1.0 and 1.0. When the match rating is typically above 0.5, we accept the hypothesis that a face exists in the region. Otherwise, we reject the hypothesis.

When a skin region contains more than three features, some of the features could be due to noise or other factors. A face model is matched to all combinations of three features, and the combination producing the highest match rating is taken as the best match for a face. Again, when the highest match rating is below 0.5, the region is considered not to contain a face. In our experiments, when a match-rating threshold of 0.5 was used, 13% of the faces were missed, while 8.7% of the detected faces were wrong.

If there are fewer than three features in a region, the models (front-view, left-view, and right-view) are matched to the region in all possible ways, and the case producing the highest match rating is taken as the best match. If the match rating for the best match is below 0.5, the region is considered not to contain a face.

Figure 4a shows the best-match pose of model 3b when matching the luminance component of Fig. 2a. The cross-correlation coefficient in this matching was 0.57. Our program locates a face in a color image by placing a square around the detected face. The square is centered at the tip of the nose of the model best matching the image.

The faces that were missed by our algorithm were due to the following reasons: 1) The faces were too small, or were broken into small pieces due to facial hair and glasses, or, were partly covered by the hat or other objects. 2) Two faces were too close to each other mering into one, or the faces were merged with hands. 3) The faces showed neither the frontal view nor the side view of persons but rather something between the two. Therefore, template matching produced low a match rating.

The detected faces that were wrong often fell in the background areas that had the color of the skin or fell in parts of the skin other than the face. As the match rating threshold is decreased, the number of missed faces decreases, but this will increase the probability of detecting false faces. We are currently investigating ways of decreasing both the false positive and false negative probabilities by using more face models in the matching and fine tuning the segmentation and feature selection modules in our algorithm.

5. Further Results and Discussion

Due to space limitations, further results are not shown here. They can, however, be found by visiting: http://www.cs.wright.edu/people/faculty/agoshtas/facechroma.html Examples of color images containing faces of different sizes and racial origins are provided. Most persons in the images have light skins, but there are persons with dark skins also. The likelihoods of pixels belonging to light-skin regions are, therefore, higher than the likelihoods of pixels in dark-skin regions. Dark-skin regions possess smaller likelihoods than light-skin regions, but their likelihoods are still considerably higher than those of nonskin regions.

We observe variations of skin likelihoods within a skin region. These variations are due to 1) skin specularity, reflecting incoming light with very little absorption; 2) change in color of skin across a region; 3) color degradation due to image compression. Many of the images we used were obtained from different websites in JPEG format. The lossy nature of JPEG compression introduces significant errors. The compression errors not only produce block-like structures in segmented images, they distort the chroma of facial features, making determination of the boundaries of features within skin regions and their sizes and orientations inaccurate.

In our experiments, no skin region was missed, although some nonskin regions were classified as skin regions. This can be attributed to the fact that our chroma chart contained information about skin colors of many races. Therefore, objects in a scene having colors similar to those of the skin were wrongly classified as the skin. The skin regions were further processed, and some were rejected when a low match rating was obtained in template matching. To locate faces in skin regions, features in the regions were used to generate hypotheses about the faces, and the existence of faces were verified by template matching.

6. Summary and Conclusions

Indexing and retrieval of images containing human activities require the ability to detect humans and, in particular, human faces in images. In this paper a new method for detecting human faces in color images was presented that first determines the skin-color regions and then determines faces within those regions. A chroma chart was prepared through a learning process that contains the likelihoods of different colors representing the skin. This chroma chart was then used to distinguish skin regions from nonskin regions.

The contributions of this paper are 1) transforming a color image into a gray scale image, with the gray value at a pixel representing the likelihood of the pixel belonging to the skin; 2) designing an automatic thresholding method to segment a skin-likelihood image and extract skin regions in an image; 3) developing a rule-based system to identify faces in skin regions. The skin-likelihood image obtained in this manner can be used to guide a face-detection algorithm and speed up the process by avoiding an exhaustive search. The automatic threshold selection method described for iso-

lating skin regions in a likelihood image can, in general, be used to segment gray scale images that require multiple of threshold values to extract homogeneous regions.

Acknowledgements

The authors would like to thank David Kriegman and Peter Belhumeur of Yale University and Bernard Achermann of the University of Bern for making their face databases available to us. The support of the National Science Foundation under grants IRI-9529045 and IRI-9509253 is also greatly appreciated.

References

[1] Qian Chen, H. Wu, and M. Yachida, "Face detection by fuzzy pattern matching," *Proc. 5th Int. Conf. Computer Vision,* 1995, pp. 591–596.

[2] A. J. Colmenarez and T. S. Huang, "Face detection with information-based maximum discrimination," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition,* 1997, pp. 782–787.

[3] T. F. Cootes and C. J. Taylor, "Locating faces using statistical feature detectors," *Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition,* 1996, pp. 204–209.

[4] J. L. Crowley and F. Berard, "Multi-modal tracking of faces for video communications," *Proc. Computer Vision and Pattern Recognition,* Puerto Rico, June 1997, pp. 640–645.

[5] Y. Dai and Y. Nakano, "Face-texture model based on SGLD and its applications in face detection in a color scene," *Pattern Recognition,* vol. 29, no. 6, 1996, pp. 1007–1017.

[6] Y. Dai, Y. Nakano, H. Miyao, "Extraction of facial images from a complex background using SGLD matrices," *Proc. 12th Int. Conf. Pattern Recognition,* vol. 1, 1994, pp. 137–141.

[7] Y. Gong and M. Sakauchi, "Detection of regions matching specified chromatic features," *Computer Vision and Image Understanding,* vol. 61, no. 2, 1995, pp. 263–269.

[8] V. Govindaraju, "Locating human faces in photographs," *Int J. Computer Vision,* vol.19, no. 2, 1996, pp. 129–146.

[9] R. M. Haralick, "Texture features for image classification," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 3, no. 6, 1973, pp. 610–621.

[10] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision,* McGraw Hill, 1995, pp. 32–35.

[11] J. K. Kasson and W. Plouffe, "An analysis of selected computer interchange color spaces," *ACM Trans. Graphics,* vol. 11, no. 4, 1992, pp. 373–405

[12] T. K. Leung, M. C. Burl, and P. Perona, "Finding faces in cluttered scenes using random labeled graph matching," *Proc. 5th Int. Conf. Computer Vision,* 1995, pp. 637–644.

[13] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada, "Facial pattern detection and color correction from television picture and newspaper printing," *Journal of Imaging Technology,* vol.16, no. 5. 1990, pp. 165–169.

[14] H. A. Rowley, S. Bluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol 20, no. 1, 1998, pp 23–38.

[15] B. Schiele and A. Waibel, "Gaze tracking based on face-color," presented in *Int'l Workshop on Face and Gesture Recognition,* Zurich, July 1995 (unpublished proceedings).

[16] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," *Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition,* 1996, pp. 236–241.

[17] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, 1998, pp. 39–51.

[18] K. C. Yow and R. Cipolla, "Finding initial estimation of human face location," *Proc. 2nd Asian Conf. Computer Vision,* vol. 3, Singapore, 1995, pp. 514–518.

[19] K. C. Yow and R. Cipolla, "A probabilistic framework for perceptual grouping of features for human face detection," *Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition,* Vermont, USA, 1996.

[20] K. C. Yow and R. Cipolla, "Feature-based human face detection," *Image and Vision Computing,* vol. 15, 1997, pp. 713–735.

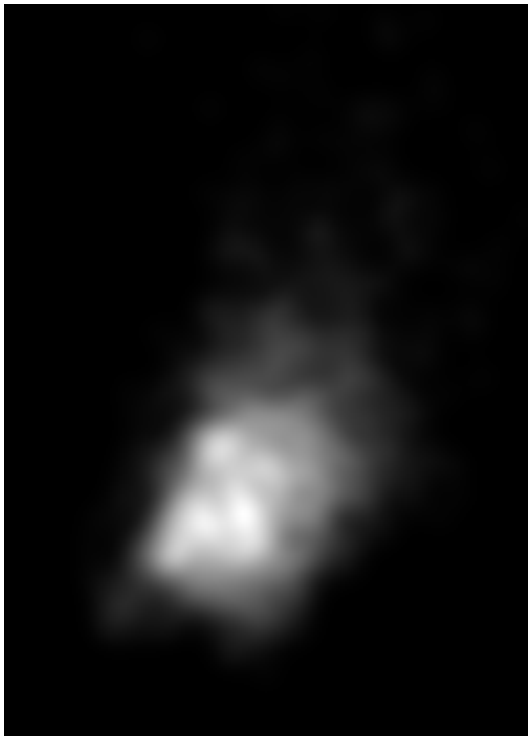

Figure 1. The chroma chart showing the likelihoods of different colors representing the skin. The horizontal and vertical axes show the a and b components of colors in the CIE LAB color space. This chroma chart is used to transform color images to skin-likelihood images.

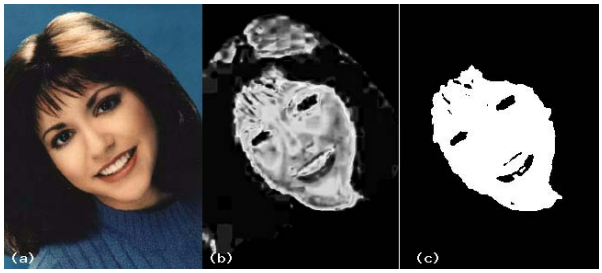

Figure 2. (a) A color image containing a face. (b) The skin-likelihood image. (c) The skin region obtained by segmenting the skin-likelihood image. Regions within this skin region are hypothesized as facial features and used in an image-to-model matching process.

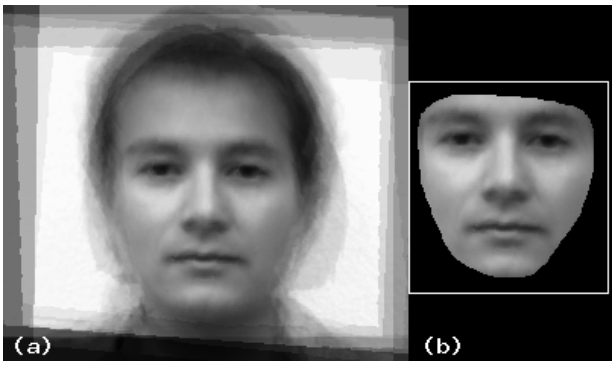

Figure 3. (a) Average of sixteen frontal-view faces of males and females wearing no glasses and having no facial hair. (b) The model face used in the template matching process to verify the existence of faces in skin regions. Left and right borders of this template are positioned at the center of the left and right ears of the averaged face shown in (a). The template is vertically centered at the tip of the nose of the model.

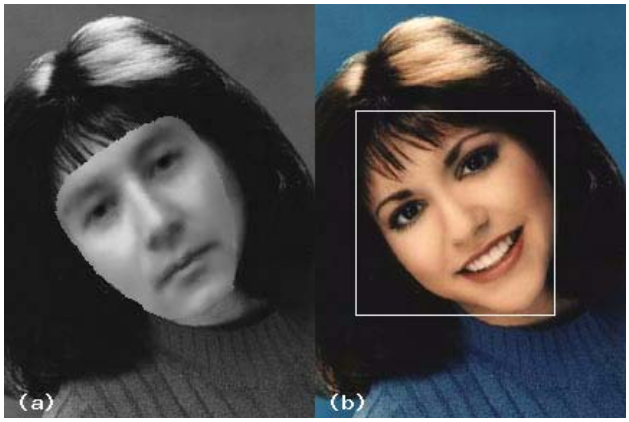

Figure 4. (a) Best-match pose of the model face shown in Fig. 3b in the luminance component of color image shown in Fig. 2a. (b) Detected face is enclosed in a square. The center of the square is at the tip of the nose of the model face when matching the luminance image.