

Face detection and recognition in color images

Smt. M.P.Satone

Associate Professor, K.K.Wagh Institute of Engineering Education And Reaserch Centre
Nashik, Maharashtra, 422003

Dr. G.K.Kharate

Principal, Matoshri College of Engineering and Research Centre,
Nashik, Maharashtra, 422003

Abstract

In this paper we describe an algorithm for a color based technique used to detect frontal human faces in images where they appear. A process for face detection, involves template matching, region clustering and color segmentation, works with high accuracy, and gives good statistical results with training images. Given the generality of the images and the templates used, the assumption would be that the implementation works well on other images, regardless of the scene lighting, size of faces or type of faces in the pictures. After detecting faces Principle component analysis is used to recognize the face of particular person from the image.

Keywords: *Face Detection, Skin Color Classification, Thresholding, Feature Extraction.*

1. Introduction

Human face perception is currently an active research area in the computer vision community. Human face localization and detection is often the first step in applications such as video surveillance, human computer interface, face recognition and image database management. Locating and tracking human faces are a prerequisite for face recognition and/or facial expressions analysis; although it is often assumed that a normalized face image is available. In order to locate a human face, the system needs to capture an image using a camera and a frame-grabber to process the image, search the image for important features and then use these features to determine the location of the face. For detecting face there are various algorithms including skin color based algorithms. Color is an important feature of human faces. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features. Under certain lighting conditions, color is orientation invariant. This property makes motion estimation much easier because only a translation model is needed for motion estimation. However, color is not a physical phenomenon; it is a perceptual phenomenon that is related to the spectral characteristics of electromagnetic radiation in the visible wavelengths striking the retina. Tracking human faces using color as a feature has several problems like the color

representation of a face obtained by a camera is influenced by many factors (ambient light, object movement, etc.).

Different cameras produce significantly different color values even for the same person under the same lighting conditions and skin color differs from person to person. In order to use color as a feature for face tracking, we have to solve these problems. It is also robust towards changes in orientation and scaling and can tolerate occlusion well. A disadvantage of the color cue is its sensitivity to illumination color changes and, especially in the case of RGB, sensitivity to illumination intensity. One way to increase tolerance toward intensity changes in images is to transform the RGB image into a color space whose intensity and chromaticity are separate and use only chromaticity part for detection. In this paper we have presented a method for face detection which has two image processing steps. First, we separate skin regions from non-skin regions. After that, we locate the frontal human face(s) within the skin regions. In the first step, we get a chroma chart that shows likelihoods of skin colors. This chroma chart is used to generate a gray scale image from the original color image. This image has the property that the gray value at a pixel shows the likelihood of that pixel of representing the skin. We segment the gray scale image to separate skin regions from non skin regions. The luminance component itself is used then, to determine if a given skin region represents a frontal human face or not. To recognize particular face from the image, PCA [7] is used. The purpose of PCA is to reduce the large dimensionality of the data space into the smaller intrinsic dimensionality of feature space.

2. Skin color model

In order to segment human skin regions from non-skin regions based on color, we need a reliable skin color model that is adaptable to people of different skin colors and to different lighting conditions [1]. The common RGB representation of color images is not suitable for characterizing skin-color. In the RGB space, the triple component (r, g, b) represents not only color but also *luminance*. Luminance may vary across a person's face due to the ambient lighting and is not a reliable measure in

separating skin from non-skin region [2]. Luminance can be removed from the color representation in the chromatic color space. Chromatic colors [3], also known as "pure" colors in the absence of luminance, are defined by a normalization process shown below:

$$r = R/(R+G+B) \quad (1)$$

$$b = B/(R+G+B) \quad (2)$$

Note: Color green is redundant after the normalization because $r+g+b = 1$.

Chromatic colors have been effectively used to segment color images in many applications [4]. It is also well suited in this case to segment skin regions from non-skin regions. The color distribution of skin colors of different people was found to be clustered in a small area of the chromatic color space. Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin colors of different people are very close, but they differ mainly in intensities [1]. With this finding, we could proceed to develop a skin-color model in the chromatic color space.

Twenty color images were used to determine the color distribution of human skin in chromatic color space. As the skin samples were extracted from color images, the skin samples were filtered using a low-pass filter to reduce the effect of noise in the samples. The impulse response of the low-pass filter is given by:

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (3)$$

Fig. 1 shows the color distribution of these skin samples in the chromatic color space. The color histogram revealed that the distribution of skin-color of different people is clustered in the chromatic color space and a skin color distribution can be represented by a Gaussian model $N(m, C)$.

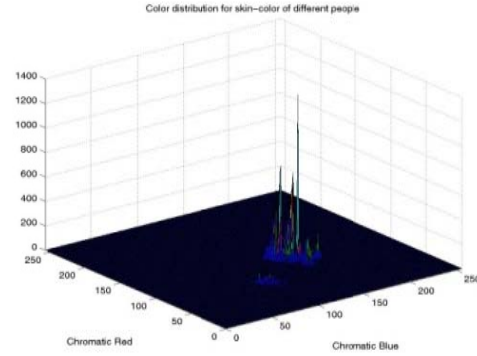


Fig. 1. Color distribution for skin-color of different people.

$$\text{Mean: } m = E \{ x \} \text{ where } x = (r \ b)^T \quad (4)$$

$$\text{Covariance: } C = E \{ (x-m) (x-m)^T \}. \quad (5)$$

Fig. 2 shows the Gaussian Distribution $N(m, C)$ fitted by our data.

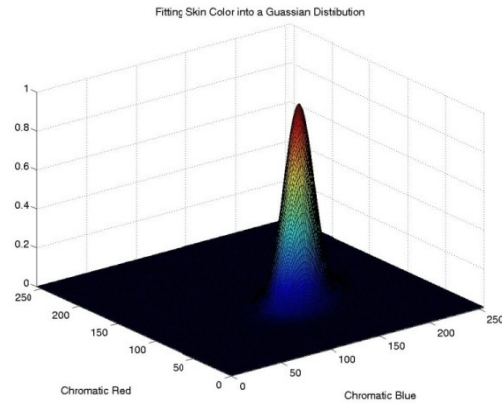


Fig. 2 Fitting skin color into a Gaussian distribution

With this Gaussian fitted skin color model, we can now obtain the likelihood of skin for any pixel of an image. Therefore, if a pixel, having transform from RGB color space to chromatic color space, and has a chromatic pair value of (r, b) , the likelihood of skin for this pixel can then be computed as follows:

$$\begin{aligned} \text{Likelihood} &= P(r, b) \\ &= \exp[-0.5(x - m)^T C^{-1} (x - m)] \end{aligned}$$

$$\text{where } x = (r, b)^T \quad (6)$$

Hence, this skin color model can transform a color image into a gray scale image such that the gray value at each pixel shows the likelihood of the pixel belonging to the skin. With appropriate thresholding, the gray scale images

can then be further transformed to a binary image showing skin regions and non-skin regions.

3. Skin Segmentation

Beginning with a color image, the first stage is to transform it to a skin-likelihood image. This involves transforming every pixel from RGB representation to chroma representation and determining the likelihood value based on the equation given in the previous section. The skin-likelihood image will be a gray-scale image whose gray values represent the likelihood of the pixel belonging to skin. However, it is important to note that the detected regions may not be necessarily corresponding to skin. It is only reliable to conclude that the detected region has the same color as that of the skin. The important point here is that this process can reliably point out regions that do not have the color of the skin and such regions would not need to be considered anymore in the face finding process. Since the skin regions are brighter than the other parts of the images, the skin regions can be segmented from the rest of the image through a thresholding process. To process different images of different people with different skin, a fixed threshold value is not possible to be found. Since people with different skins have different likelihood, an *adaptive thresholding* process is required to achieve the optimal threshold value for each run. In our program, the threshold value is decremented from 0.65 to 0.05 in steps of 0.1. If the minimum increase occurs when the threshold value was changed from 0.45 to 0.35, then the optimal threshold will be taken as 0.4. Using this technique of adaptive thresholding, many images yield good results; the skin-colored regions are effectively segmented from the non-skin colored regions. Not all detected skin regions contain faces. Some correspond to the hands and arms and other exposed part of the body, while some corresponds to objects with colors similar to those of the skin. The second stage of face finder will employ facial features to locate the face in all these skin-like segments.

3.1 Skin region

Using the result from the previous section, we proceed to determine which regions can possibly determine a frontal human face. To do so, we need to determine the number of *skin regions* in the image. A skin region is defined as a closed region in the image, which can have 0,1 or more holes inside it. Its color boundary is represented by pixels with value 1 for binary images. We can also think about it as a set of connected components within an image [2]. All holes in a binary image have pixel value of zero (black). The process of determining how many regions we have in a binary image is by labeling such regions. A label is an integer value. We used an 8-connected neighborhood in order to determine the labeling of a pixel. If any of the neighbors had a label,

we label the current pixel with that label. If not, then we use a new label. At the end, we count the number of labels and this will be the number of regions in the segmented image. To separate each of the regions, we scan through the one we are looking for and we create a new image that will have ones in the positions where the label we are searching occurs. The others are set to zero. After this, we iterate through each of the regions found in order to determine if the region might suggest a frontal human face or not.

3.2 Number of holes inside the region

Face region should have at least one hole inside that region. Therefore, we get rid of those regions that have no holes. To determine the number of holes inside a region, we compute the Euler number [5] of the region, defined as:

$$E = C - H \quad (7)$$

Where E: is the Euler number. C: The number of connected components. H: The number of holes in a region.

The development tool (Matlab) provides a way to compute the Euler number. For our case, we already set the number of connected components (i.e. the skin region) to 1 since we are considering 1 skin region at a time. The number of holes is, then:

$$H = 1 - E \quad (8)$$

where H: The number of holes in a region
E: The Euler number.

Once the system has determined that a skin region has more than one hole inside the region, we proceed to analyze some characteristics in that particular region. We first create a new image with that particular region only. The rest is set to black.

3.3 Center of the mass

To study the region, we first need to determine its area and center of the region. There are many ways to do this. One efficient way is to compute the center of mass (i.e., centroid) of the region [5]. The center of area in binary images is the same as the center of the mass and it is computed as shown below:

$$x = \frac{1}{A} \sum_{i=1}^m \sum_{j=1}^m jB[i, j] \quad (9)$$

$$y = \frac{1}{A} \sum_{i=1}^m \sum_{j=1}^m iB[i, j] \quad (10)$$

Where: B is the matrix of size [n x m] representation of the region. A is the area in pixels of the region.

3.4 Orientation

Most of the faces we considered are vertically oriented. However, some of them have a little inclination. One way to determine a unique orientation is by elongating the object. One way to determine a unique orientation is by elongating the object. The orientation of the axis of elongation will determine the orientation of the region. In this axis we will find that the inertia should be the minimum. The axis will be computed by finding the line for which the sum of the squared distances between region points and the line is minimum. In other words, we compute the least-squares of a line to the region points in the image [5]. At the end of the process, the angle of inclination (theta) is given by:

$$\theta = \frac{1}{2} \operatorname{atan} \frac{b}{a-c} \quad (11)$$

Where

$$a = \sum_{i=1}^m \sum_{j=1}^m (x' ij)^2 B[i, j] \quad (12)$$

$$b = 2 \sum_{i=1}^m \sum_{j=1}^m x' ij x' ij B[i, j] \quad (13)$$

$$c = \sum_{i=1}^m \sum_{j=1}^m (y' ij)^2 B[i, j] \quad (14)$$

And

$$x' = x - \bar{x} \quad (15)$$

$$y' = y - \bar{y} \quad (16)$$

3.5 Width and height of the region

At this point, we have the center of the region and its inclination. First, we fill out the holes that the region might have. This is to avoid problems when we encounter holes. We now proceed to determine the height and width by moving 4 pointers: one from the left, right, top and bottom of the image. If we find a pixel value different from 0, we stop and this is the coordinate of a boundary. When we have the 4 values, we compute the height by subtracting the bottom and top values and the width by subtracting the right and the left values.

3.6 Region ratio

We can use the width and the height of the region to improve our decision process. The height to width ratio of the human faces is around 1. In order to have less misses

however, we determined that a minimum good value is 0.8. Ratio values below 0.8 do not suggest a face since human faces are oriented vertically. The ratio should also have an upper limit. We determined by analyzing the results in our experiments that a good upper limit should be around 1.6. There are some situations however, that we indeed have a human face, but the ratio is higher. This happens when the person has no shirt or is dressed in such a way that part of the neck and below is uncovered. In order to account for these cases, we set the ratio to be 1.6 and eliminate the region below the corresponding height to this ratio. While the above improves the classification, it can also be a drawback for cases such as the arms that are very long. If the skin region for the arms has holes near the top, this might yield into a false classification.

3.7 Template matching

One of the most important characteristics of this method is that it uses a human face template to take the final decision of determining if a skin region represents a face. Ready template was chosen for testing. The template we used shown in Figure 3.



Fig 3. Template

Notice that the left and right borders of the template are located at the center of the left and right ears of the averaged faces. The template is also vertically centered at the tip of the nose of the model. Then compute the cross-correlation value between the part of the image corresponding to the skin region and the template face properly processed and centered. A good threshold value for classifying a region as a face is if the resulting autocorrelation value is greater than 0.6. After the system decided that the skin region correspond to a frontal human face, we get a new image with a hole exactly the size and shape of that of the processed template face. We then invert the pixel values of this image to generate a new one, which, multiplied by the original grayscale image, will yield an image as the original one, but with the template face located in the selected skin region. We finally get the coordinates of the part of the image that has the template face. With these coordinates, we draw a rectangle in the original color image. This is the output of the system. Result of face image with detection of faces is shown in following figures.



Fig. 4 The skin likelihood image Fig. 5 Segmented image

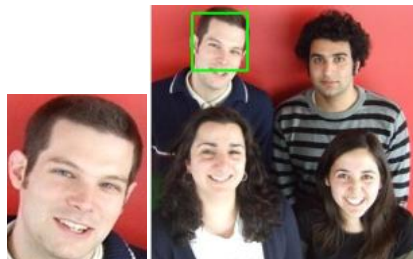


Fig. 8 Test image Fig. 9 Recognized face

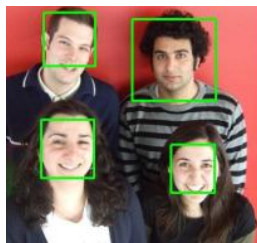


Fig. 6 Final Detection

4. Recognition of face

Training set consists of the faces detected from the image. Thus the task of the face recognition is to find the most similar feature vector among the training set to the feature vector of a given test image. Let Ω_A be the training image of a person A which has a pixel resolution of $M \times N$. In order to extract PCA [7] features of Ω_A , first image is converted into a pixel vector Ω_A by concatenating each of the M rows into a single vector. The length of the vector Φ_A will be $M \times N$. PCA algorithm is used as a dimensionality reduction technique which transforms the vector Φ_A to a vector ω_A which has dimensionality d where $d \ll M \times N$. For each training image Ω_i , we stored these feature vectors ω_i . In recognition phase, for a given test image Ω_j of a person, α_j be the identity of that person. As in the training phase, we have computed the feature vector of this person using PCA and obtain ω_j . In order to identify Ω_j , we should compute the similarities between j and the entire feature vectors ω_i 's in the training set. The similarity between feature vectors is computed using Euclidean distance. The identity of the most similar ω_i will be the output of face recognition. If $i=j$, it means that we have correctly identified the person j , otherwise if $i \neq j$, it means that we have misclassified the person j .

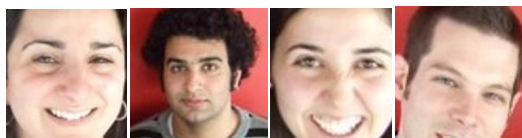


Fig. 7 Faces detected from the image and saved as training set

5. Result

The retrieval of images containing human faces requires detection of human faces in such images and then recognizing the face. We implemented a method that segments skin regions out and locate. We used 30 images to test the performance of this implementation and we got 78% of accuracy. The misses usually included regions with a similar skin likelihood values and regions that certainly were skin regions, but corresponds to other parts of the body such as arm and legs.

References

- [1] Jie Yang and Alex Waibel, "A Real-Time Face Tracker", CMUCS Technical Report.
- [2] J. Cai & A. Goshtasby & C. Yu, "Detecting Human Faces in Color Images ", Wright State University, U. of Illinois.
- [3] G. Wyszecki and W.S. Styles. Color Science: Concepts and Methods, Quantitative Data and Formulae, second edition, John Wiley & Sons, New York 1982.
- [4] Gong and M. Sakauchi, "Detection of regions matching specified chromatic features", Computer Vision and Image Understanding, vol. 61, no. 2, 1995, pp 263 – 269.
- [5] R. Ramesh, Kasturi R. and Schunck B., Machine Vision, pp 31 -51, McGraw Hill, New York 1995.
- [6] Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa3, Richa Singh," A Robust Skin Color Based Face Detection Algorithm", Tamkang Journal of Science and Engineering, Vol. 6, No. 4, pp. 227-234 (2003).
- [7] M.A.Turkand. A.P.Pentland," Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.