# Human Face Detection in Cluttered Color Images Using Skin Color and Edge Information

K. Sandeep and A.N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology – Madras
Chennai – 600 036, India
sandeepkanumuri@yahoo.co.in
raju@ee.iitm.ernet.in

## Abstract

*In this paper, we address the problem of face detection in still images. We propose a fast algorithm for detecting human faces in color images. The algorithm uses color histogram for skin (in the HSV space) in conjunction with edge information to quickly locate faces in a given image. The proposed algorithm has been tested on various real images and its performance is found to be quite satisfactory.*

## 1. Introduction

Face detection is the essential front end of any face recognition system, which locates and segregates face regions from cluttered images, either obtained from video or still image. It also has numerous applications in areas like surveillance and security control systems, content-based image retrieval, video conferencing and intelligent human-computer interfaces. Most of the current face recognition systems presume that faces are readily available for processing. However, in reality, we do not get images with just faces. We need a system, which will detect, locate and segregate faces in cluttered images, so that these segregated faces can be given as input to face recognition systems. Given an image, the goal of a face detection algorithm is to identify the location and scale of all the faces in the image. The task of face detection is so trivial for the human brain, yet it still remains a challenging and difficult problem to enable a computer to do face detection. This is because the human face changes with respect to internal factors like facial expression, beard and mustache, glasses etc and it is also affected by external factors like scale, lightning conditions, contrast between face and background and orientation of the face.

A lot of research is going on in the area of human face detection at present [3, 8]. Many researchers have proposed different methods addressing the problem of face detection. In a recent survey, Hjelmas [2] classifies different techniques used in face detection as feature-based and image-based. The feature-based techniques use edge information, skin color, motion, symmetry measures, feature analysis, snakes, deformable templates and point distribution models (PDMs). Image-based Techniques include neural networks [1], linear subspace methods like eigen faces, fisher faces, eigen faces with background learning [6] and statistical approaches like support vector machines, higher order statistics [5]. The problem of face detection in still images is more challenging and difficult when compared to the problem of face detection in video since motion information can lead to probable regions where a face could be located.

In prior studies, different human skin colors from different races have been found to fall in a compact region in color spaces [7, 9]. One can detect skin by making use of this compactness. However, there are limitations to using just the skin color. For example, when faces are too close, the skin regions can get merged. We propose the idea of using connectivity of skin and edge information to segregate faces that are close and also to separate background from faces. The advantages of the proposed approach are demonstrated on real images.

## 2. Face Dectection

Our face detection system consists of three steps. The first step is to classify each pixel in the given image as a skin pixel or a non-skin pixel. The second step is to identify different skin regions in the skin detected image by using connectivity analysis. The last step is to decide whether each of the skin regions identified is a face or not. This is done using two parameters. They are the height to width ratio of the skin region and the percentage of skin in the rectangle defined by the height and width.

### 2.1 Skin Pixel Classification

Different color spaces used in skin detection previously include HSV, normalized RGB, YCrCb, YIQ and CIELAB.

According to Zarit et al. [10], HSV gives the best performance for skin pixel detection. We conducted our own experiments independently and converged to the same fact. Our experiments also showed the superiority of HSV color space over RGB and YCrCb color spaces. Skin color can be modeled using a histogram, a single Gaussian or a mixture of Gaussians. Jones and Rehg [4] have demonstrated that the histogram model is superior to other models. Also, the histogram is easy to implement and is computationally efficient.

In the HSV space, H stands for hue component, which describes the shade of the color, S stands for saturation component, which describes how pure the hue (color) is while V stands for value component, which describes the brightness. The removal of V component takes care of varying lighting conditions. H varies from 0 to 1 on a circular scale i.e. the colors represented by H=0 and H=1 are the same. S varies from 0 to 1, 1 representing 100 percent purity of the color. H and S scales are partitioned into 100 levels and the color histogram is formed using H and S.

In order to train for skin color, we downloaded color images containing human faces from the Internet and extracted the skin regions in these images manually. Our training set contained more than 4,50,000 skin pixels to form the color histogram in H and S. For each pixel, H and S values are found and the bin corresponding to these H and S values in the histogram is incremented by 1. After the training is completed, the histogram is normalized. The histogram obtained can be seen in 3 dimensions and 2 dimensions in Fig 1(a) and Fig 1(b), respectively. Though it appears that there are two different regions very much apart with high skin probability, they both belong to the same region, since H is cyclic in nature. Also, we can see that the skin color falls into a very small region in the entire HS space. The height of a bin in the histogram is proportional to the probability that the color represented by the bin is a skin color. So we use a threshold between 0 and 1 to classify any pixel as a skin pixel or a non-skin pixel. If the threshold value is high, all the non-skin pixels will be classified correctly but some of the skin pixels will be classified as non-skin pixels. If the threshold value is low, all the skin pixels will be classified correctly whereas some of the non-skin pixels will be classified as skin pixels. This represents a trade-off between percentage of skin pixels detected as skin and percentage of non-skin pixels falsely detected as skin. There is an optimum threshold value with which one can detect most of the skin pixels and reject most of the non-skin pixels. This threshold is found by experimentation.

Given an image, each pixel in the image is classified as skin or non-skin using color information. The histogram is normalized and if the height of the bin corresponding to the H and S values of a pixel exceeds a threshold called skinthreshold (obtained empirically), then that pixel is considered a skin pixel. Otherwise the pixel is considered a non-skin pixel. A skin detected image is one in which only the skin pixels are shown. A general image and its skin detected image can be seen in Fig 2(a) and Fig 2(b), respectively.

## 2.2 Connectivity Analysis

Using the skin detected image, one knows whether a pixel is a skin pixel or not, but cannot say anything about whether a pixel belongs to a face or not. One cannot say anything about it at the pixel level. We need to go to a higher level and so we need to categorize the skin pixels into different groups so that they will represent something meaningful as a group, for example a face, a hand etc. Since we have to form meaningful groups of pixels, it makes sense to group pixels that are connected to each other geometrically. We group the skin pixels in the image based on a 8-connected neighborhood i.e. if a skin pixel has got another skin pixel in any of its 8 neighboring places, then both the pixels belong to the same region. At this stage, we have different regions and we have to classify each of these regions as a human face or not. This is done by finding the centroid, height and width of the region as well as the percentage of skin in the rectangular area defined by the above parameters. The centroid is found by the average of the coordinates of all the pixels in that region. For finding height

- The y-coordinate of the centroid is subtracted from the y-coordinates of all pixels in the region.
- Find the average of all the positive y-coordinates and negative y-coordinates separately.
- Add the absolute values of both the averages and multiply by 2. This gives the average height of the region.

Average width can be found similarly by using x-coordinates. Since the height to width ratio of human faces falls within a small range on the real axis, using this parameter along with percentage of skin in a region, the algorithm should be able to throw away most of non face skin regions. So if the height to width ratio falls within the range of well-known golden ratio $(= (1 + \sqrt{5})/2) \pm$ tolerance and if the percentage of skin is higher than a threshold called percentagethreshold, then that region is considered a face region. The algorithm works with faces of all sizes and does not assume anything about the scale at which a face appears. Instead it gives the size of the face detected which will be useful when the faces detected are sent for further processing by a face recognition system. Fig. 3(a) shows a test image taken under natural conditions. We applied the skin color algorithm on this test image and it detects the face as shown in Fig. 3(b).

The skin color algorithm will not work well in certain cases. For example, if there are colors in the image which resemble skin but are not skin pixels, it could cause two skin regions to be recognized as one because these false skin pixels will form a connecting path between the two regions. This could be avoided by using edge information since some of the false pixels will have a high gradient since they are at the boundary of the face.

### 2.3 Using Edge Information

There are many ways to perform edge detection. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero-crossings in the second derivative of the image to find edges. Sobel, Prewitt and Roberts operators come under gradient method while Marrs-Hildreth is a Laplacian method. Among these, the Sobel operator is fast, detects edges at finest scales and has smoothing along the edge direction, which avoids noisy edges. So we used Sobel operator to get the edge image. Fig. 4(a) and Fig. 4(b) shows a normal image and its edge image respectively.

We can improve the skin color algorithm by using edge information and modifying the way we classify skin pixels. A skin pixel in addition to exceeding the skinthreshold for histogram should also have a gradient less than a certain threshold called edgethreshold. Fig. 5(a) and Fig. 5(b) shows the results of the algorithm using skin color only and the algorithm using both skin color and edge information on a test image respectively. We can see that the former algorithm captures only one face whereas the latter captures all the faces.

## 3 Experimental Results

The algorithm is applied on a wide variety of images taken under different lighting conditions and with different backgrounds. The images also have areas containing skin from other parts of the body like hands, and areas with color very similar to that of skin. These areas get classified as skin and they form skin regions accordingly. We implemented the algorithm in MATLAB 5.3 on a Pentium PIII 128MB RAM Windows NT workstation. For a $320 \times 240$ image, the total time taken by the algorithm was 11 seconds. The time taken will be much less if the algorithm is implemented on C or C++. The histogram is formed using a training set of over 4,50,000 pixels drawn from various sources on the internet. The training set contained skin pixels of people belonging to different races. A few faces used in the training are shown in Fig. 6(a) to 6(f).

## The Proposed Algorithm

**Step 1.** Convert the input RGB image ( rgb(i,j) ) into HSV image ( hsv(i,j) )

**Step 2.** Get the edge map image (edge(i,j)) from RGB image using Sobel operator.

**Step 3.** For each pixel (i,j), get the corresponding H and S values.

**Step 4.** If (colorhistogram(H,S) $>$ skinthreshold) and (edge(i,j) $<$ edgethreshold)
then skin(i,j) = 1 i.e. (i,j) is a skin pixel
else skin(i,j) = 0 i.e. (i,j) is a non-skin pixel

**Step 5.** Find the different regions in the image by implementing connectivity analysis using 8-connected neighborhood.

**Step 6.** Find height, width, and centroid for each region and percentage of skin in each region.

**Step 7.** For each region, if (height/width) or (width/height) is within the range (Goldenratio $\pm$ tolerance) and (percentage of skin $>$ percentagethreshold)
then the region is a face,
else it is not a face.

The various thresholds used in the algorithm are shown in the following table. These thresholds are arrived at after some experimentation.

| Type of threshold | Value |
|---|---|
| *SkinThreshold* | 0.1 |
| *EdgeThreshold* | 125 |
| *PercentageThreshold* | 55 |
| *Tolerance* | 0.65 |

The various stages in the algorithm is explained using the boy image (Fig. 7). The initial part of the algorithm classifies each pixel as a skin pixel or a non-skin pixel. Fig. 7(b) shows the skin detected image. Fig. 7(c) shows the edge image got by using Sobel operator. The remaining part of the algorithm uses the skin detected image and the edge image, finds the skin regions and checks the height to width ratio and percentage of skin in that region. For regions classified as faces, it uses the height and width of the region to draw a rectangular box with the region's centroid as its center. The final result of the algorithm is shown in Fig. 7(d). Note that the face has been correctly located and almost at the right scale.

Some more representative results of the algorithm are shown in Fig. 8 and the results are self-explanatory. Fig. 8(a) shows a couple with a plain background. The algorithm detects both the faces but gives one false alarm, which is actually a part of the lady's hand. Fig. 8(d) has three people with a very cluttered background. All the three faces are detected but it gives three false alarms, two of them representing hand regions. Fig. 8(g) and Fig. 8(h) are images taken in the lab and all the faces are detected. Fig. 8(h) does not have any false alarms and detects both faces correctly.

Considering the fact that the algorithm was tested on a wide range of images with different faces, backgrounds etc., the performance of the algorithm is indeed encouraging. Most of the false alarms are hand regions that have their height to width ratio falling within the required range and percentage of skin in these regions will be above the percentagethreshold since the region is a real skin region. A good post-processing module is necessary to remove the few false alarms that occasionally occur.

## 4. Conclusion

A fast algorithm for face detection based on skin color, connectivity and edge information has been proposed. Though some false alarms occur, the overall performance of the proposed algorithm is still quite satisfactory. The algorithm is fast and can be used in real-time applications. The images on which the algorithm is tested are natural images taken under uncontrolled conditions and the algorithm does well on them. The algorithm locates faces but does not give the exact contour. Some kind of post-processing (such as extraction of eyes) will help to capture the face contour exactly for use by a face recognition system. Detecting faces that are partly occluded or that are overlapping remains a challenge to be addressed.

## References


[1] R. Feraud, O. J. Bernier, J. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:42–53, 2001.

[2] E. Hjelmas. Face detection: A Survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.

[3] R. L. Hsu, M. A. Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Trans. Pattern Analysis and Machine Intell.*, 24:696–706, 2002.

[4] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *Computer vision and Pattern Recognition*, pages 274–280, 1999.

[5] A. N. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri. Finding faces in photographs. In *Proceedings sixth IEEE Int'l Conference on Computer Vision*, pages 640–645, 1998.

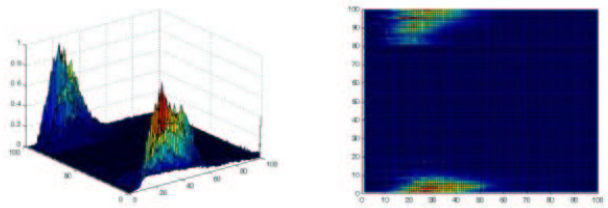
[6] R. K. Singh and A. N. Rajagopalan. Background learning for robust face recognition. In *Int'l Conference on Pattern Recognition*, 2002.

[7] J. Yang and A. Waibel. A real-time face tracker. In *Proceedings of WACV'96*, pages 142–147, 1996.

[8] M. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Analysis and Machine Intell.*, 24:34–58, 2002.

[9] T. Yoo and S. Oh. A fast algorithm for tracking human faces based on chromatic histograms. *Pattern Recognition Letters*, 20:967–978, 1999.

[10] B. D. Zarit, B. J. Super, and F. K. H. Quek. Comparison of five color models in skin pixel classification. In *ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems*, 1999.

(a) (b)

Figure 1: HS histogram for skin pixels (a) 3D view (b) 2D view

(a) (b)

Figure 2: Classification of pixels as skin or non-skin (a) Original Image (b) Skin detected Image

(a) (b)

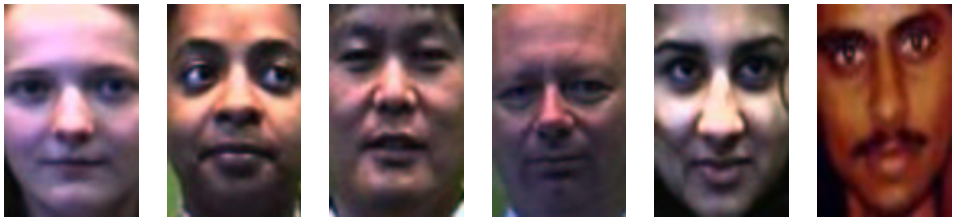
Figure 3: (a) Test Image (b) Result of skin color algorithm

(a) (b)

Figure 4: Edge detection (a) Test image (b) Edge Image

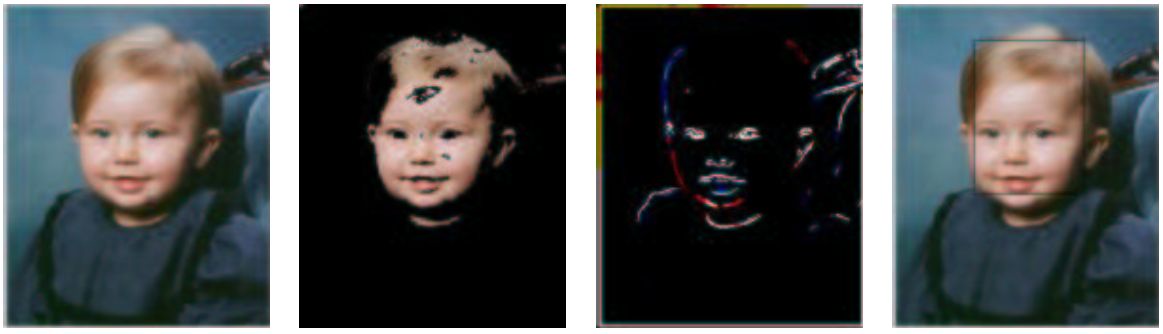
(a) (b)

Figure 5: Face detection (a) Using skin color only (b) Using both color and edge information



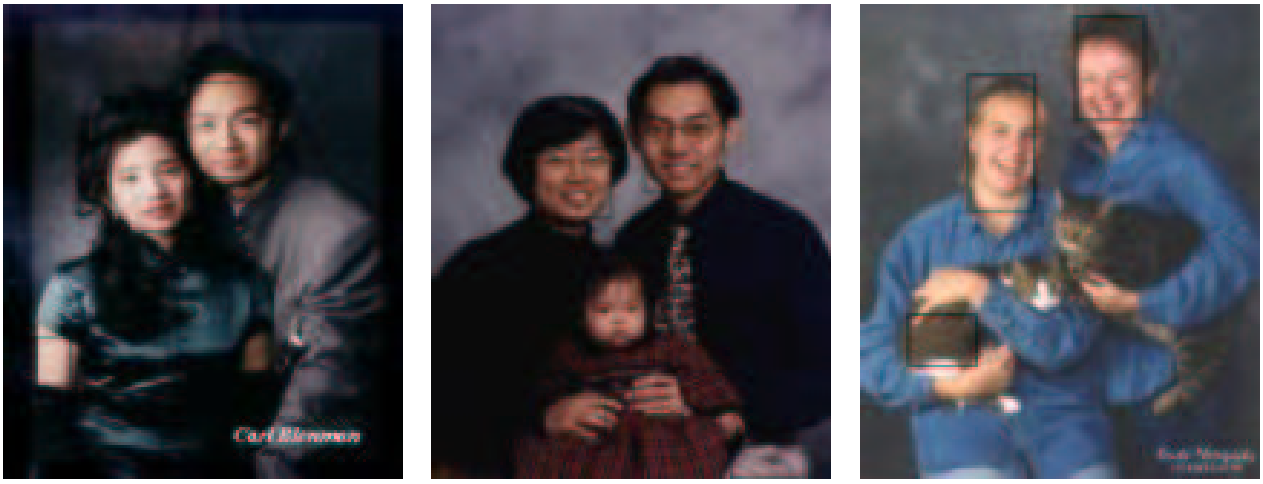
(a) (b) (c) (d) (e) (f)

Figure 6: Different training faces for skin color

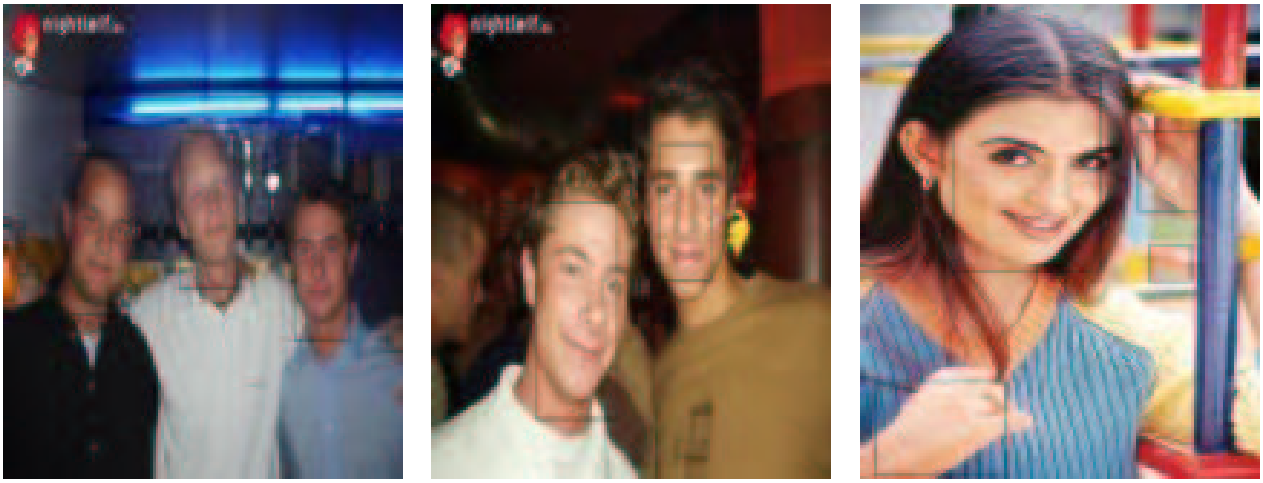


(a) (b) (c) (d)

Figure 7: (a) A test image with a boy sitting (b) Skin detected image (c) Edge image (d) Result of the proposed algorithm



(a) (b) (c)



(d) (e) (f)

(g) (h)

Figure 8: More results of the proposed algorithm