Learning Skin Pixels in Color Images Using Gaussian Mixture

Demir Gokalp

Bilkent University
Department of Computer Engineering
dgokalp@cs.bilkent.edu.tr

ABSTRACT

This paper is concerned with estimating a probability density function of human skin color using a finite Gaussian mixture model whose parameters are estimated through the EM algorithm. There are no limitations regarding if person is black or white. Two important sections of Gaussian mixture are parameter estimation and determining the number of mixture components. Experimental results show that the estimated Gaussian mixture model fits skin images from a large database. Using different color models most suitable was chosen and final outputs were showed in the experiments section.

1. INTRODUCTION

Human skin color has been used and proved to be an effective feature in many applications from human face detection to hand tracking [3]. Motivation of the using skin color information in color images is normalized skin color of different people falls into a small cluster (see Fig. 1). Generally, single Gaussian has been used for modeling the skin color distribution but this is sufficient and ineffective in many applications [5]. In this experiment, I tried to detect skin pixels in color images using Gaussian mixture model which gives more reliable result than a single Gaussian. As dataset I used news videos that have been used in TRECVID workshop series for video retrieval evaluation. There are two classes for skin and non-skin pixels. For train set 15.379 skin pixels from 42 different people whose races are different and 83.353 non-skin pixels from 102 different images are used. After training, I tested 60 different images.

First of all, to have confident solutions from skin color information, we need to normalize it. There are many kinds of techniques such as Chromatic space, HSV, LUV, YES, log-opponent [2] [7] [8]. In normalization ignoring the illumination component gives the closer values from different skin patches. In this experiment I used three different color model which are YES, chromatic space and log-opponent models. At the experiment stage I just used the E channel of the YES color model because of the minimum overlap between skin and non-skin data.

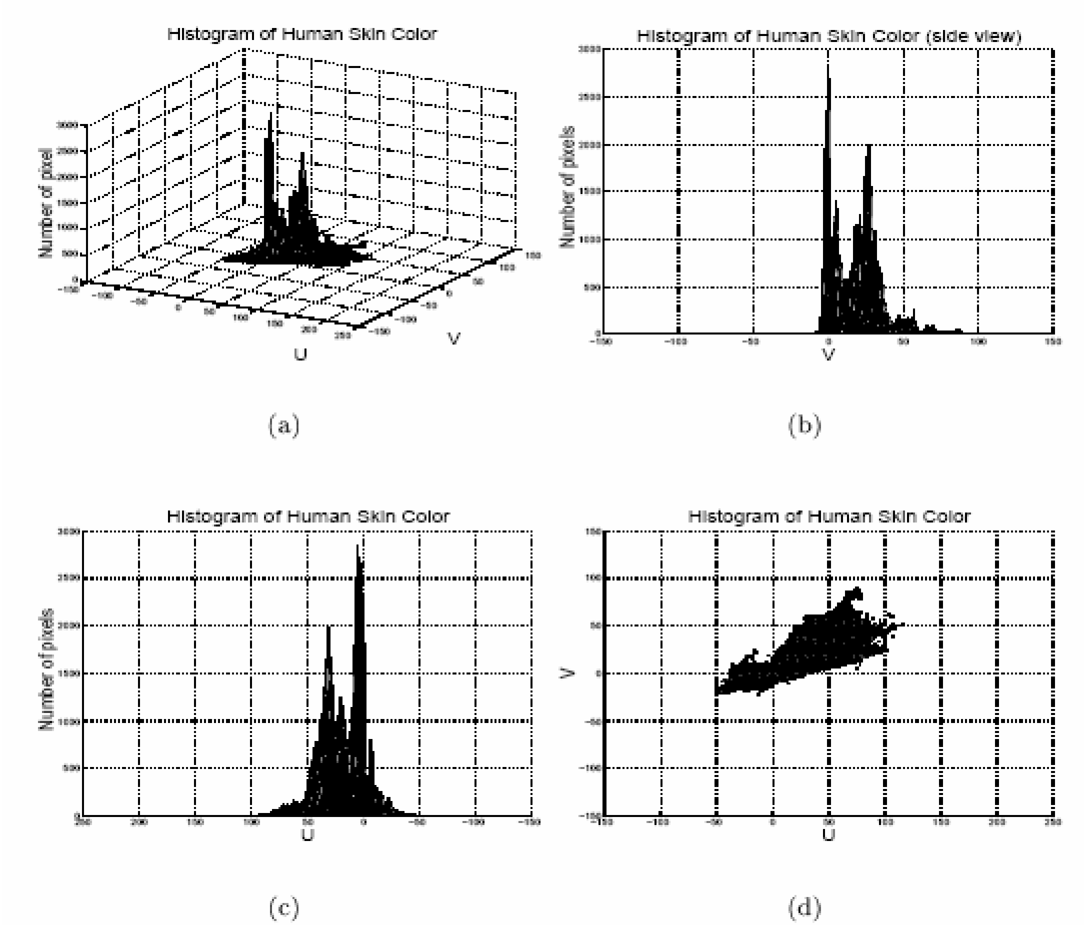


Fig. 1. Histogram of skin colors from different angles

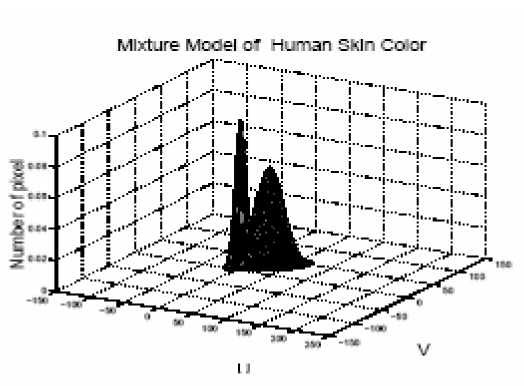
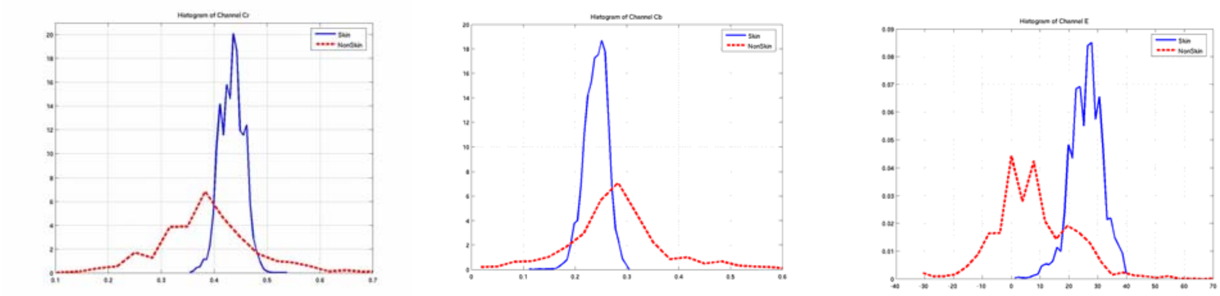
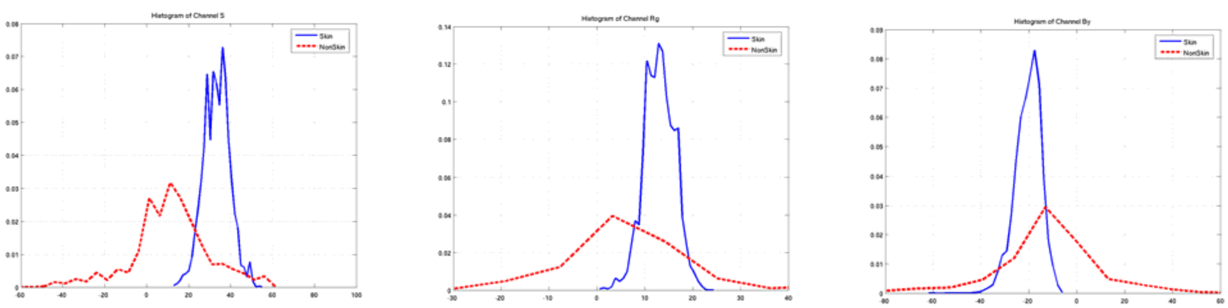


Fig. 2. Estimated density function

Using Gaussian mixtures brings two important topics which are estimation of the parameters and deciding the number of component in mixture [1]. For estimation of the parameters, I used the EM algorithm (Expectation-Maximization) whereas MDL algorithm (Minimum Description Length) is used for number of components.

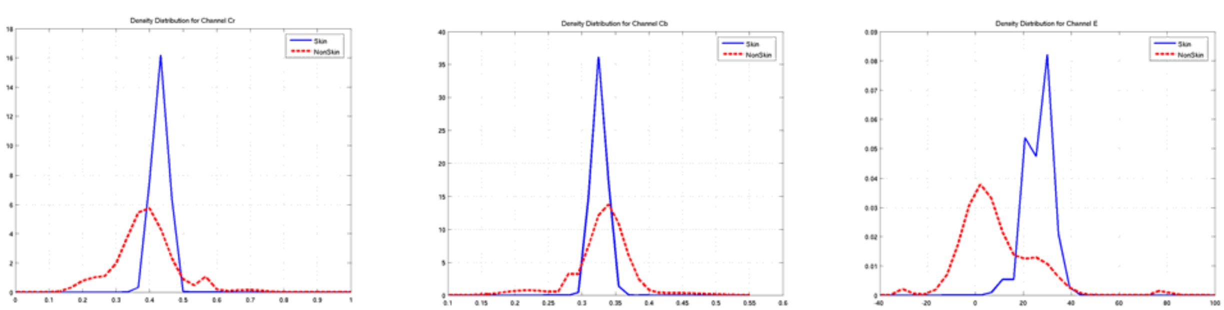


(a) Cr (b) Cb (c) E

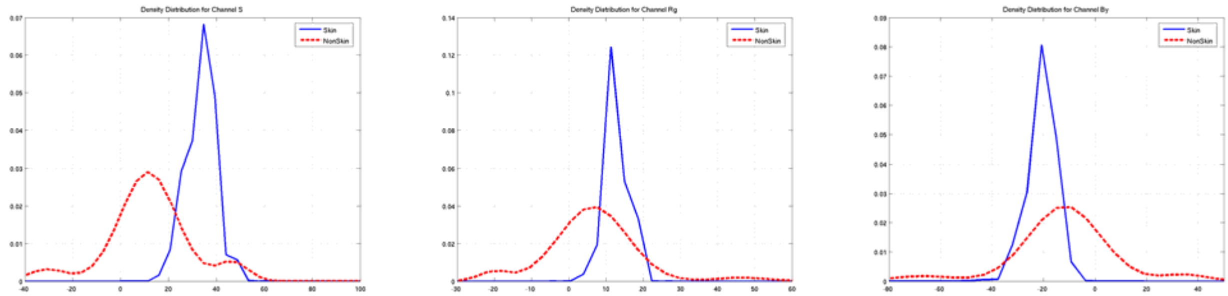


(d) S (e) Rg (f) By

Fig. 3. 1-D histograms od skin vs, non-skin pixels with respect to individual color components. Blue lines are skin, red lines are non-skin histograms.



(a) Cr (b) Cb (c) E



(d) S (e) Rg (f) By

Fig. 4. Fitted distributions for skin vs. non-skin pixels using the different color models. Blues lines are skin densities, red lines are non-skin densities.

2. MIXTURE MODEL

I will give a brief description of a finite Gaussian mixture model and then use it to model the distribution of skin and non-skin color pixels.

2.1. Gaussian Mixture Model

Both non-parametric and parametric methods are used for modeling skin color in the literature. Non-parametric techniques include piecewise linear decision boundaries such as thresholds on color features, and histogram intersection techniques that compare a control histogram computed from a skin patch given by the user and a test histogram computed from a new region in an image [4]. Parametric techniques include modeling of skin color distributions using unimodal Gaussians or mixtures of Gaussians. Nonlinear models such as multilayer perceptrons were also used. Several studies compared effectiveness of different models such as piecewise linear decision boundaries, Bayesian classifiers that use histogram-based estimation performs better than Gaussian models. However, non-parametric techniques such as histogram –based models or non linear such as multilayer perceptrons usually require a large amount of training data and computations can be quite demanding when the data size increases, so they may not be practical in many cases [6].

As can seen from Fig. 4. , the distributions of neither skin nor non-skin pixels are unimodal. To model these distributions, I use Gaussian mixture models because closed form estimates of their parameters can be computed using the Expectation Maximization (EM) algorithm. The probability density function of a mixture model with k components for the feature vector x is defined as

$$p(x) = \sum_{j=1}^{k} \alpha_j \, p(x \mid j) \tag{1}$$

where α_j is the mixture weight and $p(x \mid j)$ is the Gaussian density model for the j'th component

$$p(x \mid j) = \frac{1}{(2\pi)^{d/2} \left| \sum j \right|^{1/2}} e^{-1/2 (x-\mu_j)^T \sum j^{-1} (x-\mu_j)} \tag{2}$$

where μ_j is the mean vector and $\sum j$ is the covariance matrix for the j'th component, respectively, and d is the dimension of the feature space, $x \in \mathbb{R}^d$. I used the k-means algorithm to determine the initial configuration. Then, EM algorithm is run on the training data using a stopping criterion that checks if the change in lo-likelihood between two iterations is less than a threshold [7].

2.2. Estimating Parameters Using EM Algorithm

Various procedures have been developed to determine the parameters of a Gaussian mixture model from a set of data. Here we briefly describe the EM algorithm for parameter estimation.

For the case of Gaussian components, the mixture density contains the following adjustable parameters: π_j, μ_j, and $\sum j$.

The EM algorithm is run on the training data using a stopping criterion that checks if the change in log-likelihood between two iterations is less than a threshold. Given the training data, the parameters are estimated as [5]

$$p(j \mid x_i) = \frac{\alpha_j \, p(x_i \mid j)}{\sum_{t=1}^{k} \alpha_t \, p(x_i \mid t)} \tag{3}$$

$$\hat{\alpha}_j = \frac{\sum_{i=1}^{n} p(j \mid x_i)}{n} \tag{4}$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} p(j \mid x_i) x_i}{\sum_{i=1}^{n} p(j \mid x_i)} \tag{5}$$

$$\hat{\sum}_j = \frac{\sum_{i=1}^{n} p(j \mid x_i)(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{n} p(j \mid x_i)} \qquad (6)$$

2.3. Estimating the Number of Components

The number of components in the mixture can be either supplied by the user or chosen using some optimization criteria. I used the Minimum Description Length Principle (MDL) that tries to find a compromise between the model complexity (still having a good data approximation) and the complexity of the data approximation (while using a simple model). The purpose of statistical modeling is to discover regularities in observed data. The success in finding such regularities can be measured by the length with which the data can be described [8]. This is the rationale behind the **Minimum Description Length (MDL)** Principle introduced by **Jorma Rissanen** (Rissanen, 1978). Under MDL, the best model M is the one that minimizes the sum of the model's complexity ($\frac{KM}{2}\log n$, where KM is the number of free parameters in model M) and the efficiency of the description of the training data with respect to that model (-log $p(D \mid M)$). For a Gaussian mixture model with k components, the number of the free parameters becomes KM=(k-1) + kd + k $\frac{d(d+1)}{2}$ and the best k can be found as

$$k^* = \arg\min_k \left[\frac{KM}{2} \log n - \sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \alpha_j p(x_i \mid j)\right) \right]$$

After the densities for both skin pixels and non-skin pixels are estimated as in (1) using their corresponding training data, an unknown pixel with feature vector x is labeled as skin using the Bayesian classifier

label x as skin if p(x|skin) > p(x|non-skin)

under the minimum-error with equal priors assumption.

3. EXPERIMENTAL RESULTS

In this experiment, I manually collected image patches from skin and non-skin regions from TRECVID videos. I used 144 different color images for training data which contains 15,345 skin (in 42 images) pixels and 83,353 non-skin pixels (in 102 images). Furthermore, 60 different color images are used for testing with 13,345 skin pixels and 28,669 non-skin pixels. Test data includes skin and non-skin patches extracted independently from keyframes that are not in the training set.

Histograms for both skin and non-skin pixels in the training set are showed in fig. 3. Due to the variations in recording conditions, complexity of scene backgrounds, and the uncontrolled nature of the appearance of videos, most of the histograms for skin and non-skin have significant amount of overlap. This overlap was found to be the smallest in the E and S components of the YES model for our data.

In addition to comparing the histograms for selecting color components, I trained Bayesian classifiers using univariate Gaussians and their mixtures. The number of components in the mixture densities were estimated using Minimum Description Length formulation. For example, the optimum number of mixture components for the E color component was found to be 4. Plots for fitted densities are shown in fig. 4. confusion tables for Bayesian classification is given in Table 1. As a result, I selected the only E channel of YES model because of the small overlap in skin and non-skin histograms, and smallest classification error obtained using the Bayesian classifier.

Table 1.

		Assigned with Cr		Assigned with Cb		Assigned with Cr+Cb	
		Skin	Non-skin	Skin	Non-skin	Skin	Non-skin
TRUE	Skin	11,966	3,937	10,694	5,988	11,230	3,702
	Non-skin	1,758	24,383	3,030	22,332	2,494	24,618

(a) Classification using the normalized chromaticity color space.

		Assigned with Cr		Assigned with Cb		Assigned with Cr+Cb	
		Skin	Non-skin	Skin	Non-skin	Skin	Non-skin
TRUE	Skin	12,361	984	10,963	5,416	10,719	1,233
	Non-skin	1,363	27,336	2,761	22,904	3,005	27,087

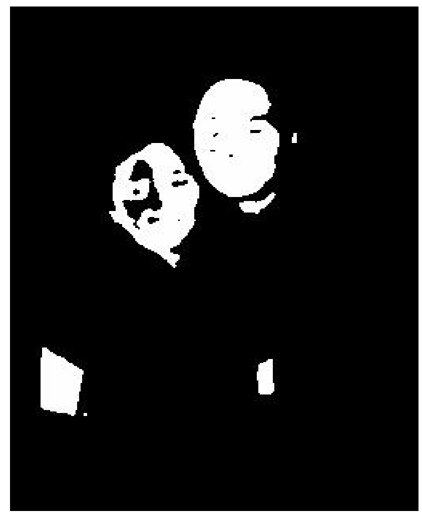
(b) Classification using the YES color space.

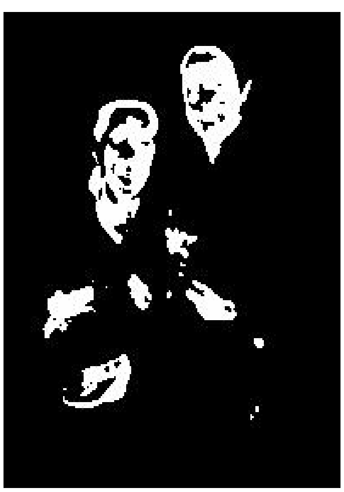
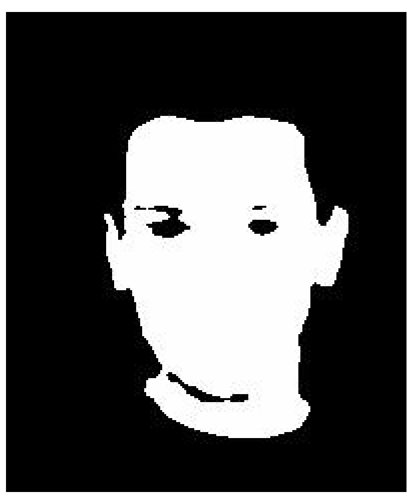
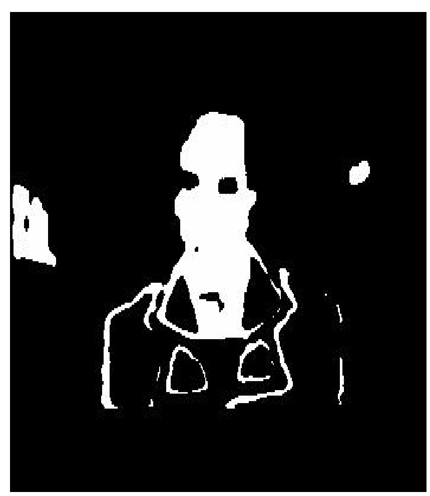
		Assigned with Cr		Assigned with Cb		Assigned with Cr+Cb	
		Skin	Non-skin	Skin	Non-skin	Skin	Non-skin
TRUE	Skin	11,873	4,265	10,355	6,164	12,004	9,191
	Non-skin	1,851	24,055	3,369	22,156	1,720	19,129

(c) Classification using the log-opponent color space.

Some example outputs of the experiment.





REFERENCES

[1] T.S. Caetano and S.D. Olabarriaga and D.A.C. Barone, "Do mixture models in chromaticity space improve skin detection", Pattren Recogniton, February, 2003.

[2] D. Saxe and R. Foulds, "Toward Robust Skin Identification in Video Images ", Proc.Second Int'l Conf. Automatic Face and Gesture Recognition, pp 379—38, 1996.

[3] J.L Crowley and F. Berard, "Multimodel Tracking of Faces for Video Communications ", Proc.IEEE Conf.Computer Vision and Pattren Recognition, pp 640—645, 1997.

[4] M.J. Jones and J.M. Rehg, "Statistical Color Models with Application to Skin Detection", Proc.IEEE Int'l Conf.Computer Vision and Pattren Recognition, pp 274--280",1999.

[5] Son Lam Phung and Abdesselam Bouzerdoum and Douglas Chai, Skin Segmentation Using Color Pixel Classification:Analysis and Comparison", IEEE Transactions on Pattern Analysis and Machine Intelligence, January, 2005.

[6] M.H.Yang and N.Ahuja, "Detecting Human Faces in Color Images", Proc.IEEE Int'l Conf.Image Processing, pp 127—130, 1998.

[7] K.Sobottka and I.Pitas, "Face Localization and Feature Extraction Based on Shape and Color Information", Proc.IEEE Int'l Conf.Image Processing, 483—486, 1996.

[8] Ming-Hsuan Yang and David J.Kriegman and Narendra Ahuja, Detecting Faces in Images:A Survey, IEEE Pattern Analysis and Machine Integlligence, January, 2002.