www.icgst.com

Adaptive Skin Color Classifier for Face Outline Models

M. Wimmer, B. Radig, M. Beetz
Informatik IX, Technische Universität München, Germany
Boltzmannstr. 3, 87548 Garching, Germany
[wimmerm, radig, beetz]@informatik.tu-muenchen.de
http://wwwradig.informatik.tu-muenchen.de

Abstract
Skin color is an important feature of faces. Applications that try to fit a face outline model to the real face contour can benefit from robust classification of skin and non-skin regions within the face. But this is a hard challenge, because skin color can look quite differently due to camera settings, illumination, shadows, people's ethnic groups etc. In this work we present a parametric skin color classifier and its adaptation to the skin color conditions within an image or image sequence. To initialize the classificatory we apply a face detector to identify a subset of pixels, which are with high probability from the skin area.
This approach can distinguish skin color from very similar colors like lip color or eye brow color. Its high speed and high accuracy makes it appropriate for real time applications such as face tracking and mimic recognition.

Keywords: *skin color classification, lip color detection, face model tracking, mimic recognition, setting low level vision parameters.*

1. Introduction
There is a widespread benefit of skin color detection in computer vision applications such as face detection, mimic recognition [2], person identification, hand gesture detection [3][8], image content filtering [4]. In this work we focus on exactly detecting skin regions and non-skin regions within human faces using a color classifier. Applications that rely on exactly fitting a face outline model, such as mimic recognition, can benefit from our approach. We built a mimic recognition systems [2] see figure 1.

Skin color occupies a subspace within the color space. The various cameras types and settings, illumination conditions, and the people's various tans and ethnic groups make that subspace very large. But not every object whose color lies within that subspace is obligatorily skin. Objects with similar color such as skin and lips can not be distinguished, often. The color of one person's lips might look like the skin of another person. The subspace of skin color intersects the subspace of lip color. But the difference of the lip color and the skin color of one person is distinct enough to distinguish between them. Figure 2 (middle column) shows some shortcomings of non-adaptive skin color classifiers: parts of the background are incorrectly classified as skin; lip color is classified as skin color, and parts of the persons' faces are incorrectly detected as non-skin color. This is especially true for the images with colored persons.

The skin color conditions within one image are mainly determined by the camera type, the camera settings, the illumination, and the person's ethnic group. Knowing the skin color conditions in advance a more specific skin color model can be used. This will improve the result of the classifier. Those skin color conditions can easily be figured out if we know some pixels to be skin color pixels. In order to acquire the skin color pixels we apply a face detector. This restricts our approach to images that include a frontal view of a person's face. Within image sequences at least one image has to include a frontal view of a face.

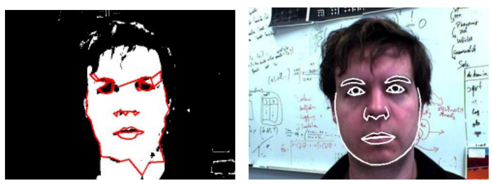

Figure 1. Left: Detecting skin color regions and fitting face model. Right: Original image with the interpolated face outline model.

The remainder of the paper is organized as follows: Section (2) gives a short overview of other techniques that focus on similar issues, section (3) exactly explains our approach, section (4) demonstrates our experimental results, and section (5) gives a resume and a short outline of future work.

2. Related work

Broadhurst and Baker [1] examine the exchange of knowledge within complex computer vision systems. Former approaches use fixed module parameters which were most often set up by hand. They introduce a framework where a high level module can feed back its knowledge to a low level module to improve the performance of the overall system. They set up a few applications, such as color based face tracking and edge based lane tracking, and compare former approaches to their approach.

Vezhnevets et al. [6] give a comprehensive overview about the work within the area of skin color detection that has been done during the last decade. They describe the main color spaces, the skin color models, and the classification techniques which they categorize into four groups. Our approach contributes to the group "dynamic skin distribution models". This group describes parametric skin color classifiers that can be adapted to the image conditions prior to classifying the image.

Rehg and Jones [4] use a common approach for estimating the distribution of skin color: Using labeled training data they created sixteen Gaussian kernels within the color space. Those Gaussians describe the probability of each pixel to be a skin color pixel. Soriano et al. [5] define their skin color model as a subspace with a distinct shape within the color space. This model can be adapted to the image illumination conditions by evaluating the image's histogram.

Viola and Jones [7] created a face detector that works with gray value images and does not use skin color at all. Their detector uses a boosted cascade of classifiers that evaluate rectangular features summing up the gray values inside. That face detector basically relies on the brighter and darker parts within faces.

3. Our technique

This section explains our approach. It can be divided into four phases. First, we locate the position and size of the face within the image. From the face area we acquire more than 100 skin color pixels. Second, we calculate the skin color condition within that image. Third we create a parametric skin color classifier. Fourth, we set up the parametric skin color classifier using the acquired skin color pixels.

3.1 Acquiring skin color pixels

In different images the color of human skin appears quite differently. The reasons are widespread and include the camera type and camera settings, the kind and position of the light source, and of course the ethnic group of the visible person. All that makes skin color detection a very hard challenge.

But within one image the camera settings, the light source, and the person's ethnic group are fixed. Therefore all skin color pixels look similar. The few remaining variances within skin color are mainly because of shadows. As an effect of that, a small number of skin color pixels can describe the condition of the skin color within the entire image. That knowledge helps to search for all the other skin color pixels within the image. Like that we can detect the exact shape of each skin color region within the image.

In order to acquire some skin color pixels we apply a face detector. That face detector must not depend on skin color, because that would falsify our expected results. We use the face detector that was proposed by Viola and Jones [7]. It returns the rough position and size of a frontal view face which is expressed as a square region around the face. We assume that there are a lot of skin color pixels within that square, so we need to figure out which ones they are. Therefore we create a mask that indicates the probability of skin color for each pixel within that square.

We previously annotated each pixel within dozens of images showing frontal faces to be a skin color or non-skin color pixel. Then we apply the face detector that delivers the squares around the faces. Each square is divided into 24x24 cells. Each cell contains the likelihood that its center pixel is one of the annotated skin color pixels. The resulting mask for skin color within the squares can be seen in figure 2. For faster application of the mask we thresholded the 24x24 grid. As a result 131 skin color pixels can be acquired very quickly.

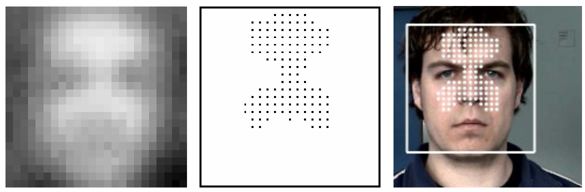

Figure 2. Left: Automatically calculated mask indicating the probability of skin color within a detected face region. Brighter color means higher probability
Middle: Discretizing and thresholding the mask for faster evaluation
Right: Application of the mask.

3.2 Calculating the skin color condition

Within the RGB color space skin color can not be modeled properly because skin color would occupy a very large subspace. Varying illumination and shadows are the reason for that. A common way to cope with that is to use the normalized RGB color space instead, see Vezhnevets et al. [6]. That color space uses the proportional part of each color, see (1). R, G, and B represent the components of the RGB color space.

$$\begin{aligned} base &= R + G + B \\ r &= R / base \\ g &= G / base \\ b &= B / base \end{aligned}$$

(1) Definition of the normalized RGB color space.

Usually b is omitted because its value can be calculated by this equation: $r + g + b = 1$

The global skin color condition are calculated within that color space using the previously detected skin pixels. We calculate the mean values (μ_r, μ_g, μ_{base}) and the standard deviation values (σ_r, σ_g, σ_{base}) of each component of the color space.

$$\begin{array}{lll} \mu_r, & \mu_g, & \mu_{base} \\ \sigma_r, & \sigma_g, & \sigma_{base} \end{array}$$

(2) Those values describe the skin color conditions within an image.

3.3 Parametric skin color classifier

Classifiers often define skin color to be situated within a subspace of a certain color space. The simplest way to describe that subspace is to define lower and upper bounds for each dimension of the color space. In the normalized RGB color space a classifier that uses a skin color subspace would look like (2). s indicates each pixel to be skin color or non-skin color. l_r, l_g, l_{base}, u_r, u_g, and u_{base} denote the lower and upper bounds of the dimensions of the color space.

$$\begin{array}{rll} s = & (r > l_r) \wedge (r < u_r) \wedge \\ & (g > l_g) \wedge (g < u_g) \wedge \\ & (base > l_{base}) \wedge (base < u_{base}) \end{array}$$

(3) Classifier that uses a skin color subspace which is defined using lower and upper bounds for each dimension within the color space.

Classifiers defining bounds are easy to use and perform at high speed but they lack robustness. The difficulty is to find proper lower and upper bounds. A large skin color subspace would correctly classify the skin of any person within any image but non-skin areas might also be classified as skin. A smaller skin color subspace would not correctly classify skin color under each circumstance.

On the one hand those bounds can be assigned with fixed values. On the other hand those bounds can be treated as parameters which can be set up in order to adapt to the current image.

For conducting our experimental results we compared our parametric skin color classifier to another classifier with fixed bounds. This is how we calculate the fixed bounds. We annotate skin color pixels and non-skin color pixels of dozens of images. Those images were taken under various illumination conditions with various camera types and settings and they show people out of various ethnic groups. Figure 3 shows some of those images. The lower and upper bounds are chosen such that the rates of both incorrectly classified skin color pixels and incorrectly classified non-skin color pixels are minimized. Those rates still remain at 10.8% for skin color and 15.3% for non-skin color. (4) shows the values for the fixed bounds.

$$\begin{array}{lcl} l_r & = & 0.36 \\ l_g & = & 0.27 \\ l_{base} & = & 280 \\ u_r & = & 0.58 \\ u_g & = & 0.36 \\ u_{base} & = & 710 \end{array}$$

(4) Fixed bounds for the non-adaptive skin color classifier.

3.4 Setting up the parametric classifier

This section shows how we adapt the parametric skin color classifier (3) to the skin color conditions of each image. Section 3.2 describes how the skin color condition (μ_r, μ_g, μ_{base}, σ_r, σ_g, σ_{base}) are acquired. Those conditions are determined by camera settings, light source, shadows, the ethnic group of the visible person and many more parameters.

Using the skin color condition we can calculate the classifier's bounds. Our first try is to restrict the skin color subspace to two times the standard deviation around the mean of the acquired skin color pixels. Thus the formulae to calculate the lower and upper bounds would look like (4).

$$\begin{array}{lcl} l_r & = & \mu_r - 2\,\sigma_r \\ u_r & = & \mu_r + 2\,\sigma_r \\ l_g & = & \mu_g - 2\,\sigma_g \\ u_g & = & \mu_g + 2\,\sigma_g \\ l_{base} & = & \mu_{base} - 2\,\sigma_{base} \\ u_{base} & = & \mu_{base} + 2\,\sigma_{base} \end{array}$$

(5) Formulae for adaptively setting up the bounds of the classifier.

Furthermore we follow another track. We calculated the lower and upper bounds for each image manually. The bounds are assigned in the way such that the rate of the incorrectly classified pixels is minimized. This can be calculated with the given ground truth for both skin and non-skin color pixels. After that we estimate a linear function for each of the six bounds that approximates those bounds with the given values of the skin color condition (μ_r, μ_g, μ_{base}, σ_r, σ_g, σ_{base}). The resulting functions can be seen in (5).

$$\begin{array}{lcl} l_r & = & 0.055 + 0.75\,(\mu_r - \sigma_r) \\ u_r & = & -0.098 + 1.385\,(\mu_r + \sigma_r) \\ l_g & = & -0.597 + 2.857\,(\mu_g - \sigma_g) \\ u_g & = & -0.17 + 1.6\,(\mu_g + \sigma_g) \\ l_{base} & = & -45.26 + 0.79\,(\mu_{base} - \sigma_{base}) \\ u_{base} & = & 765 \end{array}$$

(6) Functions for adaptively setting up the bounds of the classifier.

4. Experimental Results

4.1 Performance

Our skin color classifier must be executed in real time, because we intend to apply our classifier for face tracking, mimic recognition, and person tracking. Those three steps are executed online: (1) detect the face, (2) calculate the skin color conditions, (3) execute the classifier. The most time consuming step is (1). It can be executed in O(n) where n denotes the number of pixels within the image. This step must not be executed within each image of a sequence. Using 480x360 images this step was executed at an average of 150 ms. Step (2) uses the discretized face mask in order to calculate the mean and standard deviation values out of about 120 pixels. Executing step (2) and step (3) we achieved 107 fps evaluating a 480 x 360 image sequence and 44 fps evaluating a 640 x 480 image sequence on a 1800 MHz Pentium 4 processor. Our test program was implemented using OpenCV[1].

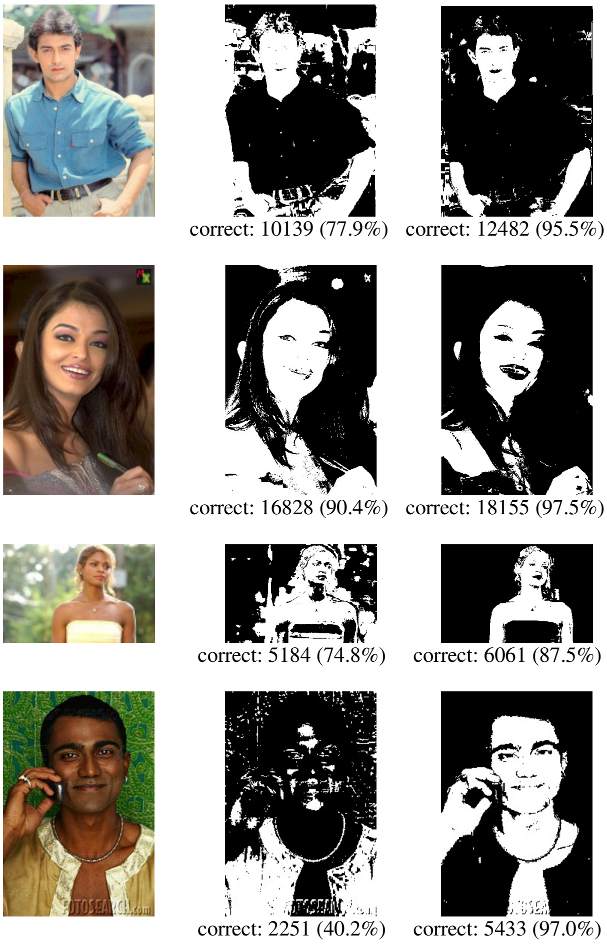

Figure 3. Comparing skin color classifiers
Left: original image,
Middle: non-adaptive classifier,
Right: adapted to the skin color condition within the image,
The numbers show the correctly classified number of pixels.

4.2 Robustness

Figure 3 shows example images of our experiments. The pictures are taken under various illumination conditions and show people out of various ethnic groups. The left column shows the original images in which each pixel was annotated to be a skin color pixel or a non-skin color pixel. The middle column shows the result of the non-adaptive skin color classifier and the right column shows the result of the adaptive skin color classifier. Note that the classifier correctly classifies facial parts such as lips and eye brows as non-skin regions. On the web there are some original images and some skin color classified images that we compared.[2]

Figure 4 shows our first attempt to classify lip color pixels. We used the same classifier but we assigned the upper and lower bounds with different formulae. Furthermore we only applied the lip color classifier within the square face region.

Figure 4. Lip color detection. The white pixels indicate the detected lip color pixels

5. Conclusion and Future Work

In different images, skin color looks differently which is a hard challenge for creating robust skin color classifiers. We propose a parametric classifier that can be adapted to each image. It classifies the skin and the non-skin regions within a human face more accurately. The parameters can be set up with previous knowledge of skin color. This knowledge is acquired using a face detector.

We demonstrated that the robustness dramatically increases compared to a non-adaptive classifier, especially in the case of poor lightning conditions and colored people. Facial parts such as eye, brows, lips, and teeth are detected correctly as non-skin color objects. Face outline models can be fit exactly to the face using that knowledge. The real time capability supports head and person tracking purposes.

We are currently working on extending our work towards classifying lip, eye brows, and other facial parts. That will improve the deformable model fitting once more (see figure 1).

6. References

[1] A. Broadhurst, S. Baker, Setting Low-Level Visioni Parameters, CMU-RI-TR-04-20, 2004

[2] S. Fischer, S. Döring, M. Wimmer, A. Krummheuer. Experiences with an Emotional Sales Agent. Workshop on Affective Dialogue Systems (ADS04) S. 309-312, Kloster Irsee, Germany, Juni 2004.

[3] S. Hämmerle, M. Wimmer, B. Radig, M. Beetz. Sensor-based Situated, Individualized, and Personalized Interaction in Smart Environments. Workshop on Situierung, Individualisierung, and Personalisierung, Informatik2005, Bonn, Germany, 19.9.2005.

[1] www.intel.com/research/mrl/research/opencv/

[2] www9.in.tum.de/people/wimmerm/skin_color_images/

[4] M. J. Jones, J. M. Rehg. Statistical color models with application to skin detection. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, 1999, 274-280.

[5] M. Soriano, S. Huovinen, B. Martinkauppi, M. Laaksonen. Skin detection in video under changing illumination conditions. In Proc. 15th International Conference on Pattern Recognition, vol. 1, 839–842. 2000.

[6] V. Vezhnevets, V. Sazonov, A. Andreeva. A Survey on Pixel-Based Skin Color Detection Techniques, Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia. 2003.

[7] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR, 2001.

[8] M. Zahedi, D. Keysers, H. Ney. Appearance-Based Recognition of Words in American Sign Language. RWTH Aachen, Germany. 2005.