Computer Vision Winter Workshop 2010, Libor Špaček, Vojtěch Franc (eds.)
Nové Hrady, Czech Republic, February 3–5
Czech Pattern Recognition Society

Weighted Skin Color Segmentation and Detection Using Graph Cuts

Rehan Ullah Khan[1], Allan Hanbury[2], and Julian Stöttinger[13]

[1] PRIP, Vienna University of Technology, Austria
[2] Information Retrieval Facility, Vienna, Austria
[3] CogVis Ltd., Vienna, Austria

Abstract *In this work we propose a principal approach combining graph cuts, color weighting and texture information to segment skin regions from images. Successfully detected faces serve as foreground seeds, no assumptions about the background is made. Further, we introduce the concept of skin weights and skin weighted images. Performance is improved by efficiently adjusting pixel weights. Prior approaches do not provide a general skin detection when the information of foreground or background seeds is not available, e.g. when there are no faces present in the image. A concept for processing arbitrary images is introduced: We learn a universal seed to overcome the probable lack of successful seed detections. Experiments on a database of 300 images with annotated pixel-level ground truth show that the scheme outperforms widely used approaches. We also evaluate the effect of different edge detectors on graph cuts based skin segmentation. It is shown that a texture and color based weighting scheme improves skin segmentation compared to a color only weighting scheme.*

1 Introduction

Skin color detection is a popular and useful technique because of its wide range of applications both in human computer interaction and content based analysis. Applications such as: detecting and tracking of human body parts [1], face detection [5], naked people detection and people retrieval in multimedia databases [6] benefit from skin detection. Also skin detection gains attention in contributing to block objectionable image or video content on the Internet automatically [31].

The most attractive properties of color based skin detection are potentially high processing speed, invariance against rotations, invariance against partial occlusions and invariance against pose changes. However, standard skin color detection techniques are affected by changing lighting conditions, complex backgrounds and surfaces having skin-like colors.

The primary objective of skin detection or classification is building a decision rule that will differentiate between skin and non-skin pixels. The most widely used approach to identifying skin colored pixels involves creating a static skin filter, a volume into which most skin pixels would fall in a given color space [32]. There is a set of techniques which estimate the distribution of skin color by a training phase. These methods are often referred to as non-parametric skin models [14]. Finally, other methods include parametric skin distribution models, such as the Gaussian skin color model [36].

This paper introduces a skin detection process using graph-cuts. The skin segmentation problem is modeled as a min-cut problem on a graph defined by the image color characteristics. The vertices of the graph represent the image colored pixels and edges represent weights or costs for labeling the vertices as skin or non-skin.

The main contribution of the paper is to use additional skin weights to augment the graph edge weights. We introduce the idea of the skin weighted image, where the pixels are replaced by their corresponding weights in the non-parametric model. The weighted non-parametric model is built from a number of different images containing skin.

The segmentation process starts by building a decision rule based on a seed. For the seed, we use faces present in the image. We therefore assume that there is a face present in the images in which we detect skin. For images without a face we use the proposed concept of a universal seed. For images that are related to each other, such as frames from the same scene in a video, it is possible to use faces found in other images as seeds in related images.

For the seed generation process we use the well known face detection approach from Viola Jones [33]. The skin segmentation technique is based on the interactive graph cuts method, first used for segmentation in [3]. There is a very efficient algorithm for finding min-cut/max-flow in a graph [2]. For general skin detection i.e. without the knowledge of the seed from image locally, we present the concept of universal seed.

The skin segmentation process is summarized in a block diagram, see Figure 1. Two types of weights have to be adjusted in the first step. One is the neighborhood weights and the other is pixel (background/foreground) weights. A graph is constructed given the proper adjustment of the neighborhood weights and background/foreground weights. A graph cut technique is used to segment the skin. Finally we evaluate the approach in the presence of different parameters.

Experiments have been performed to compare skin segmentation using (1) augmented weighted graph cuts technique (2) simple graph cuts technique (3) static models

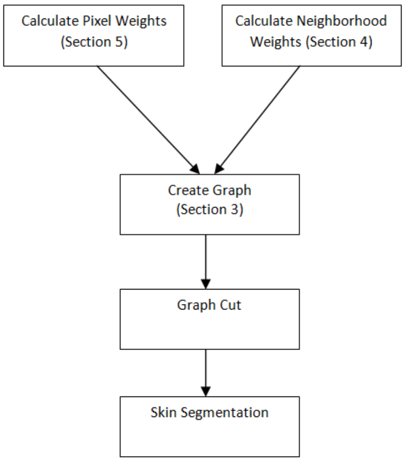

Calculate Pixel Weights (Section 5)
Calculate Neighborhood Weights (Section 4)
Create Graph (Section 3)
Graph Cut
Skin Segmentation

Figure 1: Steps for skin segmentation.

[7, 25] and (4) dynamic multiple model [16]. The results show that our approach of augmented weighting technique outperforms simple graph cuts, static and dynamic multiple models.

In Section 2 we summarize the related work regarding skin detection, usage of color spaces, graph cut approaches for segmentation and static filters for skin detection. Section 3 discusses the graph building process for skin segmentation. Section 4 describes the neighborhood weights calculation and assignment in detail. Section 5 explains the pixel weight calculation and pixel weight augmentation using skin weighted images. Experimental details and the dataset used are given in Section 6. Section 7 concludes.

2 Related Work

In computer vision, skin detection is used as a first step in face detection, e.g. [12], and for localization in the first stages of gesture tracking systems, e.g. [1]. It has also been used in the detection of naked people [8, 17] and for blocking objectionable content [31]. The latter application has been developed for videos.

The approaches to classify skin in images can be grouped into three types of skin modeling: parametric, non-parametric and explicit skin cluster definition methods. The parametric models use a Gaussian color distribution since they assume that skin can be modeled by a Gaussian probability density function [36]. Non-parametric methods estimate the skin-color from the histogram that is generated by the training data used [14].

An efficient and widely used method is the definition of classifiers that build upon the approach of skin clustering. This thresholding of different color space coordinates is used in many approaches, e.g. [26] and explicitly defines the boundaries of the skin clusters in a given color space, generally termed as static skin filters. The underlying hypothesis is that skin pixels have similar color coordinates in the chosen color space, which means that skin pixels are found within a given set of boundaries in a color space. The static filter used in YCbCr color space for skin detection as reported by [7] is,

$$\begin{aligned} Cb_{max} &= 127, Cb_{min} = 77 \\ Cr_{max} &= 173, Cr_{min} = 133 \end{aligned} \tag{1}$$

A similar static filter for RGB color space as reported by [25] is,

$$\begin{aligned} &R > 95, G > 40, B > 20 \\ &|R - G| > 15, R > G, R > B \\ &(Max\{R, G, B\} - min\{R, G, B\}) > 15 \end{aligned} \tag{2}$$

Although this approach is extremely rapid, its main drawback is a comparably high number of false detections [15]. Khan et al [16] addressed this problem by adapting the skin-color model according to reliably detected faces. When more than one detected face exists in a frame and face information is properly extracted, multiple adapted models are used. The multiple model approach makes it possible to filter out skin for multiple people with different skin tones and reduce its false positives. The dynamic multiple model approach outperforms static approaches.

Color is a low level feature that is computationally inexpensive. For many applications in computer vision, it is suitable for real-time object characterization, detection and localization [17]. The main goal of skin color detection or classification is to build a decision rule that will discriminate between skin and non-skin pixels. Following Kakumanu et al. [15] the major difficulties in skin color detection are caused by various effects such as illumination circumstances, camera characteristics, ethnicity, individual characteristics and other factors like makeup, hairstyle, glasses, sweat, and background colors. An approach for reliably detecting skin has therefore to be stable against noise, artifacts and very flexible against varying lighting conditions.

Color spaces like the HS* family model the RGB cube onto a transformed color space by following perceptual features. The angular Hue component gives the perceptive idea of a color tone. The Saturation gives a measure of the colorfulness. The HS* color spaces are broadly used in the scenarios of skin detection. Examples are found in [4, 10, 11].

To simulate the primates visual attention, perceptually uniform color spaces like the CIELAB, CIELUV are used for skin detection e.g. by [5]. Orthogonal color spaces like YCbCr, YCgCr, YIQ, YUV, YES try to form as independent components as possible. YCbCr is one of the most successful color spaces for skin detection and used by e.g. [34, 12].

Skin detection under varying illumination in image sequences is addressed in [29, 35, 30]. These approaches try to map the illuminance of the image into a common range. They compensate for the variance of changing lighting to equalize the appearance of skin color throughout different scenes. These methods are dependent heavily on the lighting correction techniques and their ability to estimate the illuminant.

Neural networks [18], Bayesian Networks e.g. [27], Gaussian classifiers e.g. [14], and self organizing maps

[4] have been used to try to increase the classification accuracy. These methods require a considerable amount of training and have high classification accuracy for a particular narrow domain.

In the literature of segmentation, Graph-cuts provide a globally optimal solution for N-dimensional segmentation when the cost function has specific properties as defined in [3]. A semi-automatic method for general image segmentation was created by Boykov et al. [3]. A user puts marks on the image, acting as a cue for being counted as segments and updating the marks without graph reconstruction. The method of Li et al. [19] consists of two steps: an object marking task as in [3] and the pre-segmentation, followed by a simple boundary editing process. The work of Shi & Malik [28] segments the image into many non-overlapping regions. They introduced normalized graph cuts and the method has often been used in combination with computing pixel neighborhood relations using brightness, color and texture cues [20, 9, 21].

Branislav et al. [24] make an assumption that each textured or colored region can be represented by a small template, called the seed and positioning of the seed across the input image gives many possible sub-segmentations of the image. A probability map assigns each pixel to just one most probable region and produces the final pyramid representing various detailed segmentations. Each sub-segmentation is obtained as the min-cut/max-flow in the graph built from the image and the seed. In our work we use detected faces as seeds for skin segmentation. Graph cuts is used for skin segmentation in [13] using both foreground and background seeds. We only use foreground seeds. Additionally we augment the weights and finally present the idea of a universal seed for general skin segmentation.

3 Graph Representing the Skin Image

The skin segmentation technique is summarized in Figure 2. The skin segmentation is based on a seed from the image on which skin segmentation is to be applied. A graph is constructed whose nodes represent pixels and whose edges represent the weights. The min-cut/max-flow algorithm presented in [2] is used for the graph cut. Later on we will show that we can perform skin segmentation without the need for a seed from the image, i.e. general skin detection.

A graph G for skin image I consists of a set of nodes V and a set of edges E that connect them. There are terminal nodes and non terminal nodes. The non-terminal nodes represent pixels in image I. The edges which connect pixels or non-terminal nodes to each other constitute a neighborhood. We call these edges $n-links$. With reference to the skin image, the edges among the non-terminal nodes represent the connectivity or neighborhood relationship between the pixels in the skin image I. The non-terminal edge weights are calculated using color and texture features and we call them neighborhood weights, explained in Section 4.

The terminal set consists of two nodes, the source S and the sink T. For the skin segmentation problem we represent these nodes as the foreground F node for "skin" and the background B node for "non-skin". These terminal nodes are connected to the non-terminal nodes through edges. The edges connecting pixels in image I and terminal nodes B and F are called $t-links$ edges. The cost of a $t-link$ connecting a pixel p and a terminal node B and F corresponds to a penalty for assigning the corresponding label to the pixel p. This is the probability value of a pixel being "skin" or "non-skin". For the skin detection problem we call it pixel weight, represent it by ω and the method for its calculation is explained in Section 5. For robust skin detection, we augment these weights using skin distribution or skin confidence where we increase or decrease the weights based on the pixel probability of being a "skin" or "non-skin" using a YCbCr skin distribution.

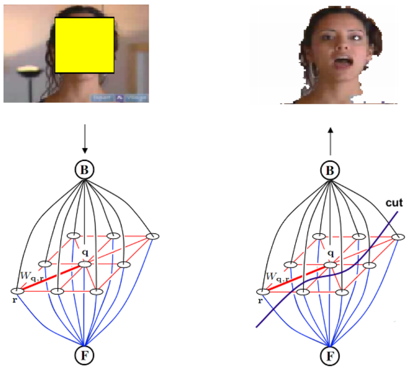
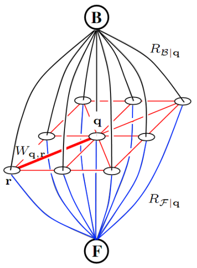


Figure 2: A graph cut for skin segmentation. A mask represents the object terminal and the whole image itself represents the background terminal. Note that hair is detected as skin because the seed covers the hair portion.

The general framework for building the graph is depicted in Figure 3. The graph is shown for a 9 pixel image and 8-point neighborhood N.


B
$R_{\mathcal{B}|\mathbf{q}}$
q
$W_{\mathbf{q},\mathbf{r}}$
r
$R_{\mathcal{F}|\mathbf{q}}$
F

edge	cost	region
$\{\mathbf{q},\mathbf{r}\}$	$W_{\mathbf{q},\mathbf{r}}$	$\{\mathbf{q},\mathbf{r}\} \in \mathcal{N}$
$\{\mathbf{q},F\}$	$\lambda\, R_{\mathcal{F}\|\mathbf{q}}$	$\forall \mathbf{q}$
$\{\mathbf{q},B\}$	$\lambda\, R_{\mathcal{B}\|\mathbf{q}}$	$\forall \mathbf{q}$

Figure 3: Left: Graph representation for 9 pixel image. Right: Table defining the costs of graph edges. λ is a constant and its value is fixed to be 1000. (source: [23]).

4 Neighborhood Weights

The non-terminal nodes represent the pixels of image I in graph G. The edges represent a neighborhood relationship between the pixels. The edge weights of neighborhood N are given by matrix $W_{q,r}$. The neighborhood size and density has great impact on the computation times. For the problem of skin segmentation we control the size and density of the neighborhood through two parameters. These parameters are window size and sampling rate. We are using a neighborhood window of size 21 × 21. For skin segmentation we are using a sampling rate of 0.3. This means that we are only selecting at random 30% sample of all the pixels in the window. There are two reasons: Firstly by using only a fraction of pixels we are reducing the computational demands and secondly only a fraction of pixels allows the use of larger windows and at the same time preserve the spatial relationship between the neighboring pixels. We show the calculation of weight matrix $W_{q,r}$ as follows.

For a black and white image the boundary penalties, as reported by [3] are

$$W_{q,r} \propto e^{-\frac{||I_q - I_r||^2}{2\sigma^2}} . \frac{1}{||q - r||} \tag{3}$$

where I_q and I_r are the intensities at point q and point r, $||q - r||$ is the distance between these points and σ is a parameter. For skin detection in color images we modify the above function to take color into account as in [22]. The straightforward modification is

$$W_{q,r} = e^{-\frac{||c_q - c_r||^2}{\sigma_1}} . \frac{1}{||q - r||} \tag{4}$$

where c_q and c_r are the RGB vectors of points at the position q and r. $||q - r||$ is the distance between these points and σ_1 is a parameter. For skin detection purpose a value of σ_1 = 0.02 is used, which is the optimized value for segmentation in [22] and is obtained experimentally for giving the best performance on a large database of images. The boundary penalty of Equation 4 only holds for images having no texture.

For taking texture into account, the neighborhood penalty of two pixels is defined as follows

$$W_{q,r} = \left(e^{-\frac{g(q,r)^2}{\sigma_2}}\right)^2 \tag{5}$$

where σ_2 is a parameter. For skin segmentation we used σ_2=0.08, and is obtained experimentally on a large database of images and

$$g(q, r) = p_b(q) + \max_{s \epsilon \mathcal{L}_{q,r}} p_b(s) \tag{6}$$

where $p_b(q)$ is the combined boundary probability and

$$\mathcal{L}_{q,r} = \{x \epsilon \mathbb{R}^2 : x = q + k(r - q), k \epsilon (0, 1)\} \tag{7}$$

is a set of points on a line from the point q (exclusive) to the point r (inclusive).

The boundary probability $p_b(q)$ involves calculation of color and texture gradients. For the final boundary calculation the color and texture gradients are combined to get a single value. To calculate the boundary probability a sigmoid function with learned parameters is used. Further details regarding boundary estimation can be found in [21].

The calculation of color gradients and texture gradients for boundary calculation is a computationally expensive operation. The alternatives for boundary calculation is using gradient based edge detectors like Sobel, Prewitt, Roberts, Gaussian, Zero-cross and Canny. The benefit of using these simple gradients lies in their low computational times compared to boundary estimation. However, though computationally beneficial, they cannot achieve the same segmentation performance as the boundary (as shown in the Section 6).

Moreover for considering computational costs we can only use color based weight function using Equation 4, where boundary calculation is not needed. However in the Experiments section we show that taking texture into account for the weight function improves skin segmentation.

5 Pixel Weights

The pixels are connected to the two terminal nodes F and B which stands for foreground and background nodes and thus we can incorporate the information provided by the automatic or manual seed/template from the skin image I. From the seed/template, we incorporate the penalty for each pixel being “skin” or “non-skin”. We use intensities of pixels which are marked as seeds to get the histogram for foreground. The histogram for background is calculated from all image pixels. These histograms are then used to set the regional penalties as negative log-likelihoods. We use the face detector introduced by Viola et al. [33] for seed selection from image I. Seed selection and its settings affect the overall skin segmentation using graph-cuts. The confidence of pixel being skin can be increased by weight augmentation. Using weight augmentation we increase the weights of a pixel by adding its weights from a skin weighted image. The constraint of having a seed patch from the local image can be softened by opting for a universal seed which is obtained from the skin samples from different faces. First we explain the regional penalty calculation based on histograms. We then explain seed selection, weight augmentation and universal seed.

The regional penalty of a point as being “skin” (foreground) $\mathcal{F}$ or “non-skin” (background) $\mathcal{B}$, as defined in [22] is

$$R_{\mathcal{F}|q} = -\ln p(\mathcal{B}|c_q) \tag{8}$$

$$R_{\mathcal{B}|q} = -\ln p(\mathcal{F}|c_q)$$

where $c_q = (c_L, c_a, c_b)^T$ stands for a vector in $\mathbb{R}^3$ of L*a*b* values at the pixel q. Here we show it for the L*a*b* color space. To compute the posterior probabilities in Equation 8 we used Bayes theorem as follows [22]

$$p_{(\mathcal{B}|c_q)} = \frac{p(c_q|\mathcal{B})p(\mathcal{B})}{p(c_q)} = \frac{p(c_q|\mathcal{B})p(\mathcal{B})}{p(\mathcal{B})p(c_q|\mathcal{B}) + p(\mathcal{F})p(c_q|\mathcal{F})} \tag{9}$$

For skin segmentation problem we first demonstrate it on $p(\mathcal{B}|c_q)$, for $p(\mathcal{F}|c_q)$ the steps are analogous. Initially we do not know a priori the probabilities $p(\mathcal{F})$ and $p(\mathcal{B})$ of the “skin” and “non-skin” regions, i.e. we do not know how large the “skin” region is compared to the “non-skin” one. Thus, we fix them to $p(\mathcal{F})$ = $p(\mathcal{B}) = 1/2$ as is also reported in [22]. After this assumption the Equation 9 reduces to

$$p_{(\mathcal{B}|c_q)} = \frac{p(c_q|\mathcal{B})}{p(c_q|\mathcal{B}) + p(c_q|\mathcal{F})} \tag{10}$$

where the "skin" and "non-skin" prior probabilities are

$$p(c_q|\mathcal{F}) = f^L_{c_L} . f^a_{c_a} . f^b_{c_b} \tag{11}$$

and

$$p(c_q|\mathcal{B}) = b^L_{c_L} . b^a_{c_a} . b^b_{c_b} \tag{12}$$

where $f_i^{\{L,a,b\}}$, resp. $b_i^{\{L,a,b\}}$, represents the foreground, resp. the background histogram of each color channel separately at the ith bin.

We smooth all histogram channels by using one-dimensional Gaussians, i.e.

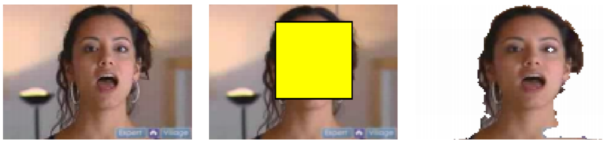
$$\bar{f}_i = \frac{1}{G} \sum\nolimits_{j=1}^{N} f_j e^{-\frac{(j-i)^2}{2\sigma^2}} \tag{13}$$

where G is the normalization factor enforcing $\sum\nolimits_{i=1}^{N} \bar{f}_i = 1$. In the skin segmentation problem, the number of histogram bins N = 64 and σ = 1. These values are obtained experimentally for giving the best performance on a large database of images. Finally $\omega = \lambda R_{\mathcal{F}|q}$, where λ is controlling the importance of penalties for foreground and background and is set to 1000 as suggested in [22].

The prior probabilities for "skin" and "non-skin" are calculated from the histograms for "skin" and "non-skin". The "skin" histogram is straight forward to compute and is computed from all the pixels in the template patch provided by the face detector. To compute the "non-skin" histogram is complex because there is no information about the background patch. It is assumed that the histogram computed from all pixels includes information on all colors (the "skin" and "non-skin") in the image I. We therefore compute the background histogram from all image pixels. Since $\sum\nolimits_{i=1}^{N} \bar{b}_i$=1, the probability $p(c_q|\mathcal{B})$ gives smaller values than $p(c_q|\mathcal{F})$ for the "skin" colors present in the template. Thus, pixels more similar to the template are assigned in the graph more strongly to the "skin" than to the "non-skin" node.

5.1 Seed/Template As Foreground

In skin image I the pixel p is connected to terminal nodes. With this setup we can incorporate the information provided by the automatic or manual seed/template from the skin image I. Manually a user can click on the image and select some small patch for skin from the image. We want to automate this process. For skin segmentation this is possible as we are using images having humans and thus probably faces visible in these images. We use the Viola Jones [33] face detector for automatic seed generation from images. The face detector finds faces and returns a rectangle around the face. The histogram for foreground is calculated from the face area. The histogram for the background is calculated from the whole image including the face area. The weights are calculated and assigned to the edges in a graph. Finally a graph cut segments the image into skin and non skin areas, as shown in Figure 4.



(a) Original image (b) Face as a seed. (c) Skin segmentation.

Figure 4: Detected face used as a seed for skin segmentation.

5.2 Pixel Weights and Skin Weights

From example images and the ground truth available for these images, we want to create a weighted skin model. This model is used to augment the pixel weights of skin graph. These weights are set on the terminal edges of the graph. These edges are connected to the source and sink nodes. The weighted model is generated in the YCbCr color space. We choose the YCbCr color space because the favorable property of this color space for skin color detection is the stable separation of luminance, chrominance, and its fast conversion from RGB. The images used to create a weighted skin model cover skin of different people having different skin tones, age, race and ethnicity group. Also images contain skin of the same person under different lighting conditions with the hope to make the model robust to skin tones and lighting conditions.

For model creation we create a 2-dimensional histogram with the Cr component of YCbCr on the x-axis, and the Cb component on the Y-axis from the manually labeled skin pixels. Given a pixel p in YCbCr color space, its Cb and Cr components are used to increment the weight at the corresponding position in the histogram. The procedure for the whole dataset results in an elliptical structure having weights for a particular Cb and Cr combination as shown in Figure 5. We want the skin detection to be independent of illumination and therefore omit the Y component.

The model in Figure 5 shows that skin covers a smaller well defined portion of the corresponding color space. The model obtained is somewhat similar in idea to the YCbCr and RGB static filters represented by Equation 1 and 2, reported in [7] and [25] respectively. The difference lies in the true representation of skin boundary. The static model constitutes a rectangular window as a filter, thus increasing the true positives, and at the same time as a result the false positives are increased. Our weighted approach reduces the false positives by creating an elliptical yet more natural and adaptive filter. This filter not only covers the true range of skin pixels but decreases false positives.

5.3 Weight Image and Pixel Weight Augmentation

We present the idea of a weight image. A weight image is created by replacing the pixel values with the weights of skin pixels. These weights are obtained from the skin weighted model in YCbCr color space. The weighted image can be used as a probability of skin. The images augment the skin detection process, where skin has to be segmented from the background. The higher the skin pixel weight the more probable that it is classified as a skin. The weighted image alone can be used to segment the skin.

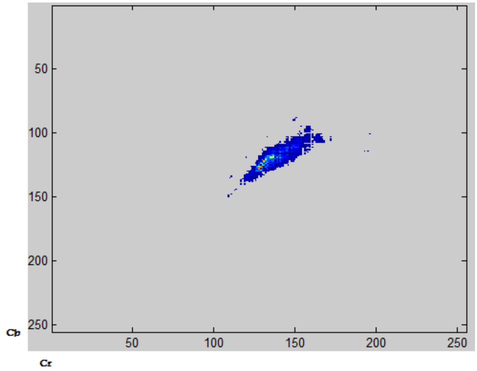


Figure 5: Skin distribution in YCbCr color space used to augment the edge weights in the directed graph.

The pixel weights are the probability weights added to the edges of the graph connected to source and sink nodes based on the seed histogram. Let's say the weight assignment procedure assigned some weight ω to the edge E of graph G, then the same pixel is found in the weighted image in the YCbCr color space. Its weight is retrieved from the weighted image and represented by w_1. Then the final new weight to be assigned to the edge E will be

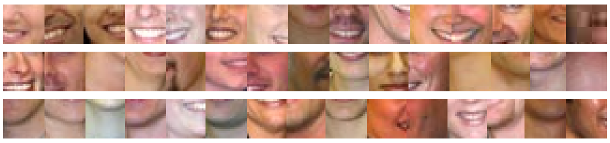
$$\omega_{new} = (\omega + w_1) \tag{14}$$

where ω_{new} is the new weight to be assigned to the edge of a graph G. We call it pixel weight augmentation. This augmentation of skin weights is only applicable to the source and sink edges and not to the neighborhood edges. The neighborhood weights or neighborhood edge weights are calculated as explained in the Section 4 and are not augmented.

5.4 From Local Seed to Universal Seed

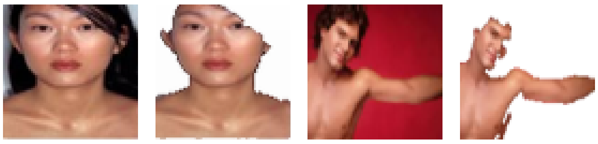
We investigate the following: Can we, without having large amount of manually labeled ground truth, produce a seed that is as general as possible and can be used as a successful filter? The seed thus required should be able to represent all kinds of skin. We define such a seed as the *universal seed*. We collect different skin tones from different faces, see Figure 6. For the universal seed calculation, different skin patches (faces) are aligned together to create one image. A face detector is used to get rectangle around the faces. The static filter is used to block the unwanted information which mainly is hair and in some cases the background information. So, a mask image is created which is pure skin patches extracted through the steps described. The foreground histogram in Equation 11 is calculated based on this mask. The histogram is saved for future use and we call such histogram as universal histogram. When a new image is to be segmented, the foreground histogram is loaded and no new histogram is calculated based on local information.

The difference of a universal seed and the skin weighted model is that universal seed is used directly to calculate weights while weighted model is used to augment these weights. The universal seed based skin detection technique is applied to images of people having different skin tones and taken under different lighting conditions and having body parts exposed. The segmentation result using the described universal seed is shown in Figure 7(b) and Figure 7(d). This concept is close to a skin filter bounded by the patches used for building the seed. The quality of skin segmentation using universal seed concept depends upon the proper selection of patches. A proper selection of skin patches improves the usability and its filtering ability. We show in the Experiments Section that the universal seed concept outperforms static filters and dynamic approaches.



(a)

Figure 6: Skin samples for the creation of a universal seed.



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

Figure 7: Skin segmentation using universal seed concept.

6 Experiments

We performed number of experiments related to skin segmentation using graph cuts. First we explain the dataset used and then we present the comparison of weighted graph cuts technique with other skin segmentation techniques using F-score. The F-score calculation involves evenly weighting recall and precision. We also show the effect of different parameters on final skin segmentation in case of graph cuts.

6.1 Data Sets

We have used images extracted from 25 videos provided by an Internet service provider that requires a skin detection application for their on-line platform. The sequences contain scenes with multiple people and/or multiple visible body parts and scene shots both indoors and outdoors, with steady or moving camera, see Figure 8. The lighting varies from natural light to directional stage lighting. Ground truth has been generated for all of the 25 videos on a per pixel basis. The data set is available one-line[1]. We also collected images from the web using Google image search. For the special case of graph cuts we have used 300 images. There are 4.69 million skin pixels and 19 million non-skin pixels to be classified.

[1] http://www.prip.tuwien.ac.at/people/julian/skin-detection

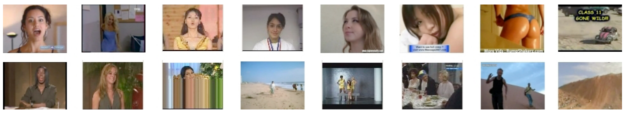


Figure 8: Example frames from the annotated video data-set.

6.2 Static, Dynamic, Graph Cut and Weighted Graph Cut Approaches

The weighted graph cuts is compared with simple graph cuts where we do not augment the weights. The addition of weights using equation 14 increases overall segmentation performance, see Figure 9. The F-Score for weighted graph cuts and simple graph cuts is calculated using the CIELAB color space. Also color plus texture is taken into account and thus the natural boundary estimation. Figure 9 shows an improved F-Score when compared with the simple graph cuts, dynamic model and static model. The static model used is from YCbCr color space which is based on [7] and the boundary values used are given by Equation 1. The dynamic model used is from [16] and is based on model adaptation using faces. We conclude that addition of weights from the weighted image increases the overall efficiency of skin segmentation system.

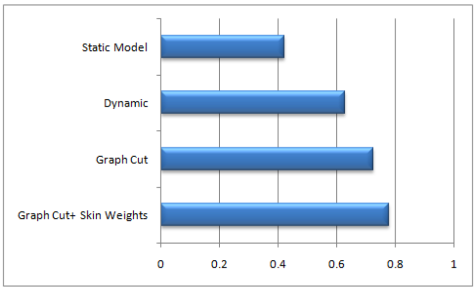
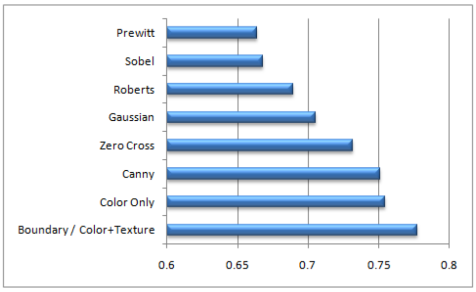


Figure 9: The F-Score for different skin segmentation techniques. Weighted Graph cuts technique outperforms simple Graph cuts (without weight augmentation), dynamic model and static model.

6.3 Effect of Different Edge Detectors

An edge detector has a profound effect on the skin segmentation using graph cuts when used instead of boundary estimation for taking color and texture into account. The effect of using different edge detectors is shown using the overall F-Score, see Figure 10. The F-Score is calculated while using the CIELAB color space. The weight augmentation technique is also used. With these parameters, the highest F-Score is related to boundary probability technique and the lowest to the Prewitt edge detector method. A drawback of using boundary probability is the time it takes during calculation. While simple edge detector is 10 times faster then the boundary probability calculation. This shows that though computationally expensive, the natural boundary estimation has high skin detection efficiency as compared to simple edge detectors.



(a)

Figure 10: Comparison of F-Score for different edge detectors and the boundary probability technique.

6.4 Effect of Color Only and Color Plus Texture

We compared the two approaches which are color only and color plus texture for weight function. Figure 10 shows the F-score for texture, color only approach and different edge detectors. The F-Score is calculated using the CIELAB color space. The weight augmentation technique is also used. The boundary estimation is used for texture calculation rather than simple edge detectors. It can be noted that color plus texture has a high F-score compared to color only. The other advantage of using color to texture plus color is the execution times. The color only approach executes 10 times faster than color plus texture, and thus can be adopted for real time detection of skin in videos.

6.5 Universal Seed

The universal seed based skin segmentation is compared with static filters in YCbCr and RGB color spaces. The YCbCr and RGB static filters are represented by Equation 1 and 2, reported in [7] and [25] respectively. On the dataset the universal seed has a higher F-Score than YCbCr and RGB static skin filters as shown in Figure 11. The results in Figure 11 for the universal seed is not as good as Figure 9, because no local seed is obtained from images and the segmentation is based on universal seed which is bounded by the samples used. As a fall back method for images without faces, this universal seed is nevertheless useful.

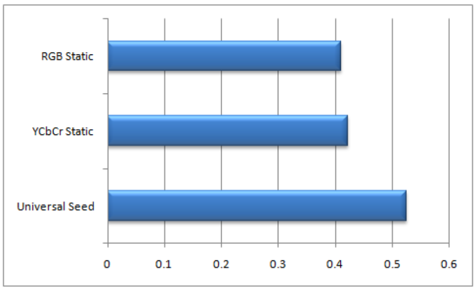


Figure 11: F-Score of universal seed and static skin filters.

7 Conclusion

We propose a principal approach combining graph cuts, color weighting and texture information to segment skin regions from images. The concept of skin weights and the skin weighted image are introduced. It is shown that we improve the overall skin segmentation performance compared to other color based approaches. For more general skin detection i.e. without the information of foreground or background seeds, we advocate the idea of a universal seed. The effect of different edge detectors on graph based skin segmentation is evaluated. We showed that the combination of texture and color based weighting scheme improves skin segmentation compared to color only approaches.

In future we plan to work on universal seed adaptation and improving the over-all execution time.

Acknowledgment

This work was partly supported by the Austrian Research Promotion Agency (FFG), project OMOR 815994, the Higher Education Commission of Pakistan under Overseas scholarships (Phase-II-Batch-I) and the CogVis[2] Ltd. However, this paper reflects only the authors views; the FFG HEC or CogVis Ltd. are not liable for any use that may be made of the information contained herein.

References

[1] Antonis A. Argyros and Manolis I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, pages 368–379, 2004.

[2] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[3] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.

[4] D. Brown, I. Craw, and J. Lewthwaite. A SOM based approach to skin detection with application in real time systems. In *BMVC'01*, pages 491–500, 2001.

[5] J. Cai and A. Goshtasby. Detecting human faces in color images. *Image and Vision Computing*, 18:63–75, 1999.

[6] Liang-Liang Cao, Xue-Long Li, Neng-Hai Yu, and Zheng-Kai Liu. Naked people retrieval based on adaboost learning. In *MLC*, pages 1133–1138, 2002.

[7] D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In *Int. Conf. Automatic Face and Gesture Recognition*, pages 124–129, 1998.

[8] Margaret M. Fleck, David A. Forsyth, and Chris Bregler. Finding naked people. In *ECCV*, pages 593–602, 1996.

[9] Charless Fowlkes, David Martin, and Jitendra Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:54, 2003.

[10] Zhouyu Fu, Jinfeng Yang, Weiming Hu, and Tieniu Tan. Mixture clustering using multidimensional histograms for skin detection. In *ICPR*, pages 549–552, Washington, DC, USA, 2004.

[11] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, 1(3):264–277, Sep 1999.

[12] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain. Face detection in color images. *PAMI*, 24:696–706, 2002.

[13] Zhilan Hu, Guijin Wang, Xinggang Lin, and Hong Yan. Skin segmentation based on graph cuts. *Science and Technology*, 14(4):478 – 486, 2009.

[14] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, 2002.

[15] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *PR*, 40(3):1106–1122, 2007.

[16] Rehanullah Khan, Julian Stöttinger, and Martin Kampel. An adaptive multiple model approach for fast content-based skin detection in on-line videos. In *ACM MM, AREA workshop*, 2008.

[17] Jiann-Shu Lee, Yung-Ming Kuo, Pau-Choo Chung, and E-Liang Chen. Naked image detection based on adaptive and extensible skin color model. *PR*, 40(8):2261–2270, 2007.

[18] Jae-Young Lee and Yoo Suk-in. An elliptical boundary model for skin color detection. In *ISST*, pages 579–584, 2002.

[19] Yin Li, Jian Sun, Chi K. Tang, and Heung Y. Shum. Lazy snapping. In *SIGGRAPH*, pages 303–308, New York, NY, USA, 2004.

[20] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[21] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[22] Branislav Micusík and Allan Hanbury. Steerable semi-automatic segmentation of textured images. In *SCIA*, pages 35–44, 2005.

[23] Branislav Micusík and Allan Hanbury. Supervised texture detection in images. In *CAIP*, pages 441–448, 2005.

[24] Branislav Micusík and Allan Hanbury. Automatic image segmentation by positioning a seed. In *ECCV (2)*, pages 468–480, 2006.

[25] Peter Peer, Jure Kovac, and Franc Solina. Human skin colour clustering for face detection. In *EUROCON*, pages 144–148, vol.2, 2003.

[26] Som Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color pixel classification: Analysis and comparison. *PAMI*, 27(1):148–154, 2005.

[27] Nicu Sebe, Ira Cohen, Thomas S. Huang, and Theo Gevers. Skin detection: A Bayesian network approach. In *ICPR*, pages 903–906, 2004.

[28] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[29] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *PAMI*, 26(7):862–877, July 2004.

[30] M. Storring, H.J. Andersen, and E. Granum. Estimation of the illuminant colour from human skin colour. *AFGR*, pages 64–69, 2000.

[31] Julian Stöttinger, Allan Hanbury, Christian Liensberger, and Rehanullah Khan. Skin paths for contextual flagging adult videos. In *ISVC*, pages 303–314, 2009.

[32] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreev. A survey on pixel-based skin color detection techniques. In *ICCGV*, pages 85–92, 2003.

[33] Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[34] K.W. Wong, K.M. Lam, and W.C. Siu. A robust scheme for live detection of human faces in color images. *SPIC*, 18(2):103–114, 2003.

[35] Jie Yang, Weier Lu, and Alex Waibel. Skin-color modeling and adaptation. In *ACCV*, pages 687–694, London, UK, 1997. Springer-Verlag.

[36] Minghsuan Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *SPIE*, pages 458–466, 1999.

[2] www.cogvis.at