Technical Report UCH-DIE-VISION-2006-01

“SKINDIFF – Robust and Fast Skin Segmentation”

Javier Ruiz-del-Solar and Rodrigo Verschae

Computational Vision Group
Department of Electrical Engineering
Universidad de Chile

March 2006

SKINDIFF – Robust and Fast Skin Segmentation

Javier Ruiz-del-Solar and Rodrigo Verschae

Dept. of Electrical Engineering, Universidad de Chile, Santiago, CHILE
E-mail: jruizd@ing.uchile.cl, rverscha@ing.uchile.cl

Abstract

Skin detection or segmentation is employed in many tasks related to the detection and tracking of humans and human-body parts. However, standard pixel-wise skin detection algorithms are not robust enough for dealing with some real-world conditions, such as changing illumination and complex backgrounds containing surfaces and objects with skin-like colors. This situation can be improved by incorporating context information in the skin detection process, as humans do. Based on this idea, \we propose a skin detection approach that uses neighborhood information (local spatial context). In this approach two main processing stages are defined, pixel-wise classification and spatial diffusion.

The proposed approach is efficiently implemented by the so-called skindiff algorithm, which achieves a robust segmentation of skin regions at a high processing speed. Skindiff is described and experimental results of its application are also presented. This includes a sensitivity analysis of the algorithm's parameters, and a comparison with state of the art skin detection algorithms, where skindiff outperforms all baseline algorithms.

Keywords: Skin Detection, Skin Segmentation, Spatial Diffusion

I. Introduction

Skin detection or segmentation is a very popular and useful technique for detecting and tracking human-body parts, specially faces and hands. Its most attractive properties are: (i) high processing speed due to its low-level nature, and (ii) invariance against rotations, partial occlusions and pose changes. However, standard skin detection techniques are not robust enough for dealing with complex environments. Changing lighting conditions and complex backgrounds containing surfaces and objects with skin-like colors are major problems that limit its use in practical real-world applications.

For solving the mentioned drawbacks, many groups have centered their research on selecting the color-space most suitable for skin detection. Many different color models have been employed, among them: RGB [20][29], normalized RGB [2][24], HIS-HSV [14][21], YCbCr [10][4], YIQ [8], YES [19], YUV [1], CIE XYZ [5], CIE LUV [29], and ab [14]. Normalized RGB and HSV are the most common used ones [22]. However, we believe that just selecting the "best" color space does not solve the mentioned drawbacks and that only marginal improvements can be achieved (see for example [22][3]).

Some authors have modeled the physical properties of the acquisition process. In [18] the physical properties and calibration of the camera to be employed in the acquisition process are modeled, while in [10] a color constancy method (white balancing) for obtaining some invariance against changing illumination is employed. In our opinion these approaches are not general enough and they can be implemented only in some special cases.

Other authors have used statistical models for solving the skin/non-skin classification problem. Most successful approaches are Mixture of Gaussians (MoG) [29][12] and histogram models [12][11]. They differ in the parametric or non-parametric form of computing the skin/non-skin probabilities. We think that statistical models are in the right direction for dealing with the real-world problems of skin detection, however they miss the benefits of using context information.

All mentioned approaches are based on the same *pixel-wise* processing paradigm, in which each image pixel is individually analyzed. We think that this paradigm should be extended; context information should be incorporated in the skin detection process. Human beings can detect skin in real scenes, or in pictures and videos without problems. However, for a human being the classification of a single pixel as skin or non-skin is a very difficult task. We believe the reason is that human skin detection is not a simple low-level process, but a process in which high-level mechanisms are also involved. If we think about the human perception of a blue ball under variable illumination, we will agree in the fact that the ball is perceived blue as a whole, and not as a ball having blue patches and some other color patches generated by differences in illumination producing highlights and shadows. In order to have this kind of perception not only low-level color processing mechanisms for blue pixels and patches detection are involved, but also shape detection mechanisms for detecting the ball circular shape and mechanisms for color constancy and interpolation [28]. In the same way, the detection of skin in a face or in a hand does not involve only low-level color processing mechanisms, but also high-level processes to assist the detection of skin (detection of hair, detection of clothes, etc), and some spatial diffusion mechanisms employed in any human segmentation process of colors and textures (cell mechanisms present in cortical area V4) [28]. Taking this fact into consideration we propose the use of context information in the skin detection process. Spatial context can be employed for skin detection in still images, while temporal context can also be incorporated when detecting skin in videos. When dealing with high-level tasks such as face and hand detection, high-level information (e.g. detected faces) can assist the skin detection process. Obviously, the skin detection process assists also these high-level detection processes. There is a kind of cooperative interaction between these low-level and high-level processes until stability in the detections is achieved. Clearly the detection and the recognition of faces are processes that can also be improved by using context information. As an example, in [16] is described how local context (image pixels surrounding the possible face areas) improves the detection of faces.

As a first step in the direction of using context in the skin detection process, we propose a robust skin segmentation approach that uses neighborhood information (local spatial context), i.e. the decision about the class (skin or non-skin) of a given pixel considers information about the pixel's neighbors. Although this idea has been employed before in the post-processing of skin-detections (e.g. holes in skin detected areas are filled using morphological dilation and isolated skin pixels are deleted using morphological erosion), in this work a different and more radical approach is employed: a diffusion process (region growing) is implemented for determining the skin pixels. The aim of this process is not just the grouping of neighbor skin pixels, but the determination of skin areas where the skinness[1] of pixels is larger than a minimal value and where the pixels' intensity between neighbors changes smoothly. Before diffusion the skinness values should be calculated. Therefore, the proposed skin segmentation approach is defined by two main stages: *pixel-wise classification* and *spatial diffusion*. The pixel-wise classification stage determines the pixel's skinness. It can be implemented using any standard skin detection algorithm applied over a given color space. Important is that the algorithm gives a skinness pixel value as output (a crisp output could be fuzzified). The spatial diffusion stage will refine the pixel-wise skin detection results, by incorporating neighborhood information for determining smooth skin segments. Therefore, using our approach, different skin segmentation algorithms can be derived.

Even though one would think that a diffusion process could slow-down the skin segmentation process, this is not the case. Thanks to the use of LUTs (look-up-tables) for implementing the pixel-wise classification and making use of a stack for avoiding recursion in the diffusion process, a high processing speed is achieved in the so-called skindiff algorithm.

This report is structured as follows. In section II some related works are outlined. In section III the proposed skin detection approach is explained and the skindiff algorithm described. In section IV experimental results of skindiff are presented. This includes a sensitivity analysis of the algorithm's

[1] Skinness is defined as the belonging of a pixel to the skin class. It corresponds to a real number between 0 and 1.

parameters and a comparison with other algorithms using real-world images. Finally, in section V some conclusions of this work are given.

II. Related Work

In a seminal paper Jones and Rehg describe the use of statistical models for solving the skin detection problem [12]. They analyze the use of MoGs and histograms for deriving skin and non-skin models. Using the obtained models they attain excellent skin detection results. Our work differs from [12] mainly in: (i) the use of LUTs for implementing the MoG statistical models, and (ii) the incorporation of local spatial context information in the skin segmentation. In this way the here proposed skin segmentation algorithm is more robust than the original Jones&Rehg's algorithm, but having about the same computational time. This is validated with the results presented in section IV, where skindiff is favorably compared with the Jones&Rehg's algorithm.

In [11] is presented a skin detection algorithm based on histogram models, whose main contribution is the use of single and adjacent pixel's skinness information for computing the histograms. The obtained segmentation accuracy is 3% superior to the situation when standard histograms are employed (Johnes&Rehg algorithm), but the computational time increases by about a factor ten. Our work differs from this mainly in the fact that instead of looking for the co-occurrence of skin pixels, we use the neighbors' information for controlling a diffusion process applied after the skin model (MoG in our case). Another important difference is that our algorithm is much faster, about ten times faster. In [27] is proposed a Bayesian Network approach for detecting skin, which also uses neighbors' information. However, in that approach only 3x3 pixels' neighborhoods are considered. Similarly, in [31] is proposed a Maximum Entropy approach that uses neighbors' information, but in this case just two neighbor pixels are considered. In [2] Albiol et al. describe a skin segmentation algorithm composed by three stages: skin detection, unsupervised segmentation, and region grouping. The unsupervised segmentation of skin is implemented in two steps, first using the chrominance channels and then using the luminance channel. Both segmentations are based on the watershed transformation. It is difficult to compare the results obtained in that work with ours, since no quantitative results are presented there (no detection rate, ROC curves or running time information is given). In any case it should be noted that our algorithm uses just one segmentation stage instead of two. In [6] is presented a face detection system that employs skin information for segmenting face regions. First, the whole image is segmented, and each segmented region is represented by the mode of its color pixels. Then, the mode is classified as skin or non-skin. If the mode is classified as skin, the whole region will be declared as a skin-color region. Similarly, in [9] is described another image segmentation system with applications to face detection, in which after segmenting the image into color-homogeneous regions, the skin segments are determined as the ones where the mean pixel's skin probability is larger than a given threshold. It is not possible to compare the results obtained in these both works with ours, because only face detection results are presented there. In [13][17][30], region-growing techniques are used to group pixels within a face or regions of a face, under controlled conditions. However, these methods are not used as in our work for detecting skin.

Some authors have used adaptive skin models for improving the skin detection rate. In [32] Zhu et al. proposed a two-step adaptive framework for skin segmentation. In the first step, a generic MoG is employed for performing a rough skin classification (high detection rate and high false positive rate). In the second step, these rough results are refined using a second MoG, trained using the Expectation-Maximization algorithm, in the reduced pixels space generated by the pixels classified as skin in the first step. In this last step, a SVM is employed for identifying the skin Gaussian (from the non-skin Gaussian), using spatial and shape features along with Gaussian parameter features. Our work differs from [32] mainly because we do not use adaptive skin models but a pre-defined skin model together with spatial context information. We believe that our approach is much faster and also more robust than the Zhu et al. approach. In section IV we show that the processing time of our approach is comparable with the one of a single MoG stage model (i.e. Jones&Rehg). Indeed by using LUTs our approach is much faster. The Zhu et al. approach is slower; in the first stage one MoG is employed, but for computing the second stage a second MoG should be used, together with a SVM. Regarding robustness, both approaches were tested using the TDSD dataset [32] (segmentation results of our approach in this dataset are shown in section IV). By comparing the segmentation results it can be seen that our approach outperforms the one of Zhu et al. in the lower part of the ROC curves, while in the upper parts both approaches achieve similar results. In [23] an approach for skin segmentation under time-varying illumination is proposed. The skin segmentation is implemented using color histograms, which are dynamically updated. This system is proposed for segmenting skin in video sequences and therefore histograms can be updated by modeling the color distribution over time. Thus, histograms are dynamically adapted based on feedback from the current segmentation and predictions of a second order discrete-time Markov model. The parameters of the Markov model are estimated using Maximum Likelihood Estimation. Although the resulting system is very robust (the obtained segmentation accuracy is 24% superior than employing standard histograms), it is not directly comparable to our approach, because it is proposed for processing video sequences, and it exploits the relative similarities and differences between the current and the previous frames.

Preliminary versions of the here-proposed approach were presented in [25][26]. This technical report is an extension of these early works, carrying out a more extensive analysis of the algorithm properties and performing a more exhaustive algorithm testing.

III. Proposed Skin Segmentation Approach

The here-proposed skin segmentation approach has two steps: (i) Pixel-Wise Classification, and (ii) Spatial Diffusion. The pixel-wise classification stage can be implemented using any standard skin detection algorithm applied over a given color space. Important is that the algorithm gives a skinness pixel value as output. The spatial diffusion stage will refine the pixel-wise skin detection results, by considering neighborhood information for determining smooth skin segments. Therefore, using our approach different skin segmentation algorithms can be derived. In this section skindiff, one these algorithms, will be described.

A. The Spatial-Range Domain

For the sake of simplicity in the description of our algorithms we will use the spatial-range domain as image space. In this domain each image pixel x_j has two parts, a spatial part (position) and a range part (intensity). The superscripts s and r denote the spatial and range parts of the pixels, respectively [15]:

$$x_j = (x_j^s, x_j^r) = (x_j^s, I(x_j^s)) \qquad (1)$$

B. Pixel-Wise Classification

Individual pixels are classified as skin or non-skin pixels in this stage. The skinness, i.e. the belonging of a pixel to the skin class, is computed as a real number between 0 and 1. Fuzzy or probabilistic approaches can be employed for implementing this task. Any color space can be used.

When a fuzzy approach is employed, the skinness value corresponds to the membership degree of the pixel to the skin fuzzy set. In this case any crisp pixel-wise detection algorithm can be employed as a basis algorithm, and its results can be then fuzzified. For instance, if an elliptical skin cluster is used as pixel-wise classification method on a given color space, before classification the cluster is fuzzified, e.g. a value “1” is assigned to the center of the cluster and decreasing values are assigned to the other positions.

When a statistical model is used, the skinness value corresponds to the probability of being a skin pixel. Two very successful methods for computing these probabilities are MoG [29] and histogram models [12], being their main difference the way in which the skin/non-skin probabilities are calculated. MoG is a parametric model and thereafter less training samples are required for obtaining good probabilities' estimators. Usually it is argued that histograms are faster because they are implemented using LUTs [12], but LUTs can also be used for implementing the MoG. In [12] slightly better detection results are obtained using histograms instead of MoG. The reason seems to be the use of an inaccurate non-skin model in the MoG case.

C. Spatial Diffusion

The final decision about the pixel's class is taken using a spatial diffusion process that takes into account spatial context information. In this process a given pixel will belong to the skin class if and only if its similarity (calculated in a given color space or even in the skinness space), with a direct diffusion-neighbor that already belongs to the skin class, is larger than a certain threshold (T_{diff}). The seeds of the diffusion process are pixels with a large skinness value, i.e. their probability of being skin or their membership degree to the skin class is larger than a given threshold (T_{seed}). The extension of the diffusion process is controlled using a third threshold (T_{min}), which defines the minimal skinness allowed for a skin pixel. The diffusion algorithm is shown in figure 1.

```
diffusion_algorithm(I, $S_{skin}$){
    find_seeds(I, $S_{seed}$)
    foreach $s \in S_{seed}$
        PushItem($S_{skin}$, s)
        diffusion_rec(s, $S_{skin}$)
}
find_seeds(I, $S_{seed}$) {
    foreach $x_i \in I$
        if $g(x_i^r) > T_{seed}$
            PushItem($S_{seed}$, $x_i$)
}
diffusion_rec(s, $S_{skin}$){
    foreach $x_j^s \in ConnectedNeighbors\ (s^s)$
      if $x_j \notin S_{skin}$
        if $g(x_j^r) > T_{\min}$
            if $d(x_j^r, s^r) < T_{diff}$
                PushItem($S_{skin}$, $x_j$)
                diffusion_rec($x_j$, $S_{skin}$)
}
```

I: input image.
S_{skin}: Final set of skin pixels, the output of the algorithm.
S_{seed}: Set of seed pixels.
PushItem(*S*,*x*): Adds *x* to the end of *S*.
ConnectedNeighbors(*s*) returns the 8-connected neighbors of *s* in the 2D lattice of the image.
g(x): Pixel-wise skin detection function. It returns a skinness value.
d(x,y): Distance function that corresponds to a dissimilarity measure.

Figure 1. Diffusion algorithm.

D. The Skindiff Algorithm

The skindiff algorithm implements the ideas described in the former two sections. Skindiff uses the RGB color space, MoG models implemented using LUTs as pixel-wise classification algorithm, and the diffusion algorithm presented in the former section. Skindiff is optimized for achieving simultaneously robustness and high processing speed. The RGB color space was selected for simplicity and speed reasons. Because of most of the public images (e.g. web images) are in this format, its use avoids a color space conversion, which could slow down the whole detection algorithm.

By looking at the pixel-wise classification and diffusion algorithms it can be noticed that both can be joined in a single processing step. If this is done, it is also not necessary to store the skin probabilities in the LUTs, but only the information concerning the following three situations: skin probability larger than T_{seed}, smaller than T_{min} or in [T_{seed}, T_{min}]. Therefore for each possible RGB combination, only 2 bits need to be stored: LUT[].seed ("1" means >T_{seed}) and LUT[].min ("1" means >T_{min}). The LUTs are filled off-line using the following procedure. First, the MoG parameters are obtained. In this case we use the same parameters employed in [12] (16 Gaussian Kernels). Second, for each RGB combination rgb_i, the mixture density function is evaluated, and LUT[rgb_i].seed and LUT[rgb_i].min are set. For reducing the size of the LUTs the RGB values are quantized. In our case we use 16, 32, 64 or 128 bins in each channel, depending on the accuracy requirement.

```
skindiff_algorithm(I, $S_{skin}$){
    find_seeds(I, $S_{seed}$)
    foreach $\tilde{s} \in S_{seed}$
        PushItem($S_{skin}$, $\tilde{s}$ )
        PushItem(Stack, $\tilde{s}$ )
        while($Stack \neq \phi$)
          $s \leftarrow$ PopItem(Stack)
          foreach $x_j^s \in ConnectedNeighbors(s^s)$
            if $x_j \notin S_{skin}$
              if LUT[$x_j^r$].min
              if $|x_j^r - s^r|^2 < T_{diff}^2$
                PushItem($S_{skin}$, $x_j$)
                PushItem(Stack, $x_j$)
}
find_seeds(I, $S_{seed}$) {
    foreach $x_i \in I$
        if LUT[$x_i^r$].seed
            PushItem($S_{seed}$, $x_i$)
}
```

I: input image.
S_{skin}: Final set of skin pixels, the output of the algorithm.
S_{seed}: Set of seed pixels.
Stack: Stack data structure.
PushItem(*S*,*x*): Adds *x* to the end of *S*.
PopItem(S): Removes and returns the element at the end of *S*.
ConnectedNeighbors(*s*): Returns the 8-connected neighbors of *s* in the 2D lattice of the image.

Figure 2. Skindiff algorithm. The diffusion algorithm is implemented using LUTs and a stack. LUT[].min and LUT[].seed are defined in the main text.

The diffusion algorithm is further optimized using an explicit stack for avoiding the algorithm's recursion. A last optimization is the use of the square of the Euclidean distance of the pixels' intensity as dissimilarity measure. This avoids the computation of the pixels' skinness when processing the images. The skindiff algorithm is shown in figure 2.

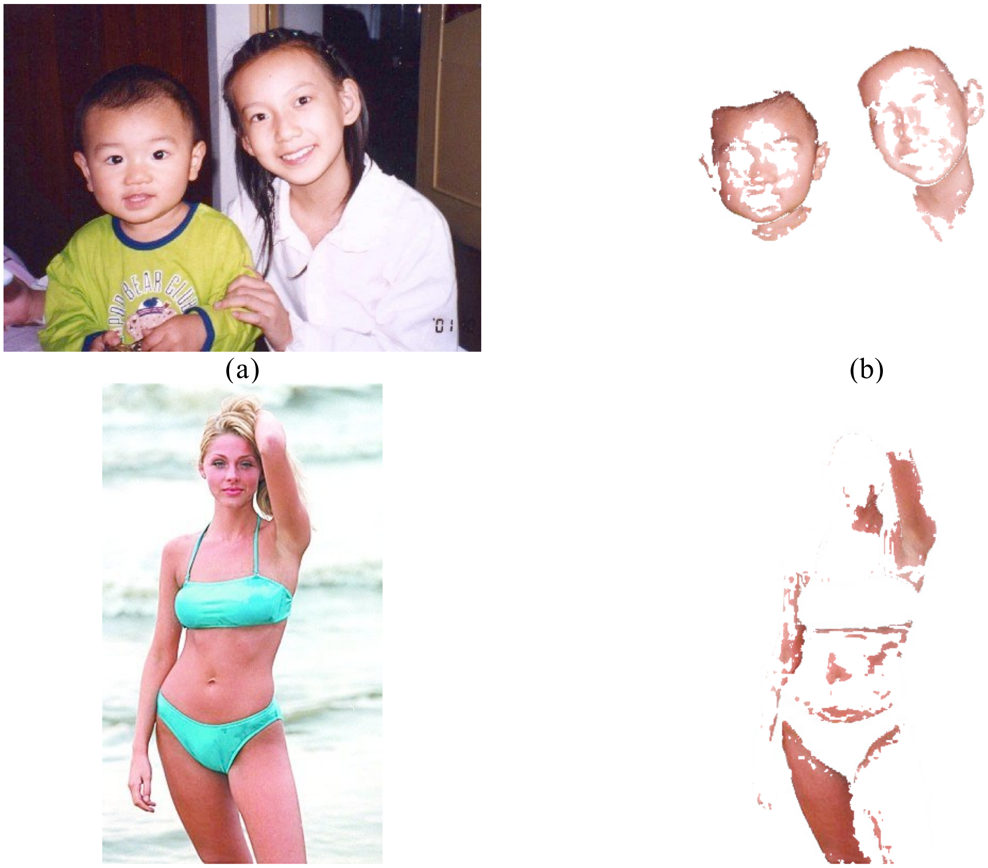
In this work we present skindiff in two different flavors, skindiff-j1 and skindiff-j2. They differ in how the pixel's skinness is computed. In the first case, the skinness is determined using the MoG skin color model proposed by Jones&Rehg. In skindiff-j2 case, it is also incorporated the MoG non-skin color model from Jones&Rehg; the skinness is determined as the normalized ratio of both models.

IV. Experimental Methodology and Results

A. Experimental Datasets

For testing the performance of skindiff in an independent dataset, we looked for large public skin detection datasets. The obvious choice was to use the Jones&Rehg's test dataset (6,818 images). However this dataset is not available anymore for public use (personal communication with the database creators). We then tried with the Sigal et al. dataset [23]. However, in this dataset *don't care* regions (boundary regions that are difficult to segment by the human operator) are employed when labeling the ground truth frames. Although this idea seems to be very useful, we do not agree with its use, mainly because in most of the skin detection situations the defined don't care regions are precisely the regions where the segmentation fails. Therefore, a skin detection algorithm should not be trained and evaluated avoiding the use of such regions, because in this way the weakness of the algorithm are hidden. Finally, we found the TDSD dataset [33], which contains 555 images (24 million skin pixels and 75 million non-skin pixels). TDSD creators consider that these images are difficult to classify, since they were selected as objectionable by a filtering software. The main problem with this dataset is that many images are very unsatisfactorily annotated (see examples in figure 3). The reason seems to be the use of a semi-automatic process for finding the ground truth information. For this reason we divided TDSD in three sets: *db-tdsd-1* that contains 100 very good annotated images (14.56 million pixels, 24.5% skin), *db-tdsd-2* that contains 254 images (45.1 million pixels, 22.2% skin), that we considered are acceptably good annotated, and finally *db-tdsd-3* that contains 201 images (42.8 million pixels, 24.3% skin), that are clearly bad annotated. The images corresponding to these three sets are listed in [34].

After several experiments we realized that for an accurate skin segmentation analysis it is very important to have images with the exact ground truth information. This is very difficult to obtain using automatic or semi-automatic annotation, hence fully human annotation should be employed. Consequently we created a new dataset composed by 101 images obtained from Internet and from digitized news videos, which was fully annotated by a human operator. The selected images are considered very difficult to segment, because they have either changing lighting conditions or complex backgrounds containing surfaces or objects with skin-like colors. This allows us not only to evaluate the performance of the different algorithms, but also to analyze the effect of the skindiff parameters' in the algorithm's performance. This dataset is called *db-skin*, and its total number of pixels is 33.1 millions, from them 5.75 millions (17.42%) correspond to skin pixels. The selected images and their ground truth information are available for future studies in [34]. Finally, for testing the behavior of the skin detection algorithms in a large set of non-skin images we selected 5,882 images (1,567 million non-skin pixels) from Internet sites and digitized news videos. This dataset is called *db-nskin*.



(a)

(b)

(c)

(d)

Figure 3. Some examples of very bad annotated images from the TDSD dataset. Image 515 (a) and its ground truth (b); Image 487 (c) and its ground truth (d).

B. Skindiff Parameter Analysis

In this section is analyzed the effect of the skindiff parameters T_{diff}, T_{min}, T_{seed}, as well as the LUTs' size, in the algorithm performance. This analysis is carried out using ROC curves, i.e. DR (detection rates) vs. FPR (false positive rates), which are built using the db-skin dataset for different LUT's quantization. From all possible operation points of the ROC curves, which are obtained using different combinations of the three algorithm's parameters, we require to select the ones that define the Pareto-Front [7], i.e. the operation points that have the largest detection rate for a given false positive rate. These points define an upper envelope of all the operation points. For this analysis we employs the skindiff-j1 flavor of skindiff.

In figure 4 is shown a *cloud* of operation points for the basic skindiff-j1 algorithm, obtained using the db-skin dataset. These operation points were obtained using the following parameter values: T_{min} in [0.04,0.4] with a step size of 0.04, T_{seed} in [0.4,1.0] with a step size of 0.04, and T_{diff} in [2,30] with a step size of 4. LUTs quantized to 16, 32, 64 and 128 bins per color channel were considered in the analysis. Figure 5 shows the corresponding Pareto-front sets. It can be noticed that when using 128 or 64 bins per channel, the best performance is obtained. Between these both quantization alternatives there is almost no difference in performance. When using 32 bins the performance decreases by about 1-2 percent or less, and when using 16 bins the performance decreases by about 3-4 percent or less, depending on the operation zone. Therefore, mainly for speed reasons, from now on LUT-64, i.e. LUTs quantized to 64 bins per channel, will be employed in this report.

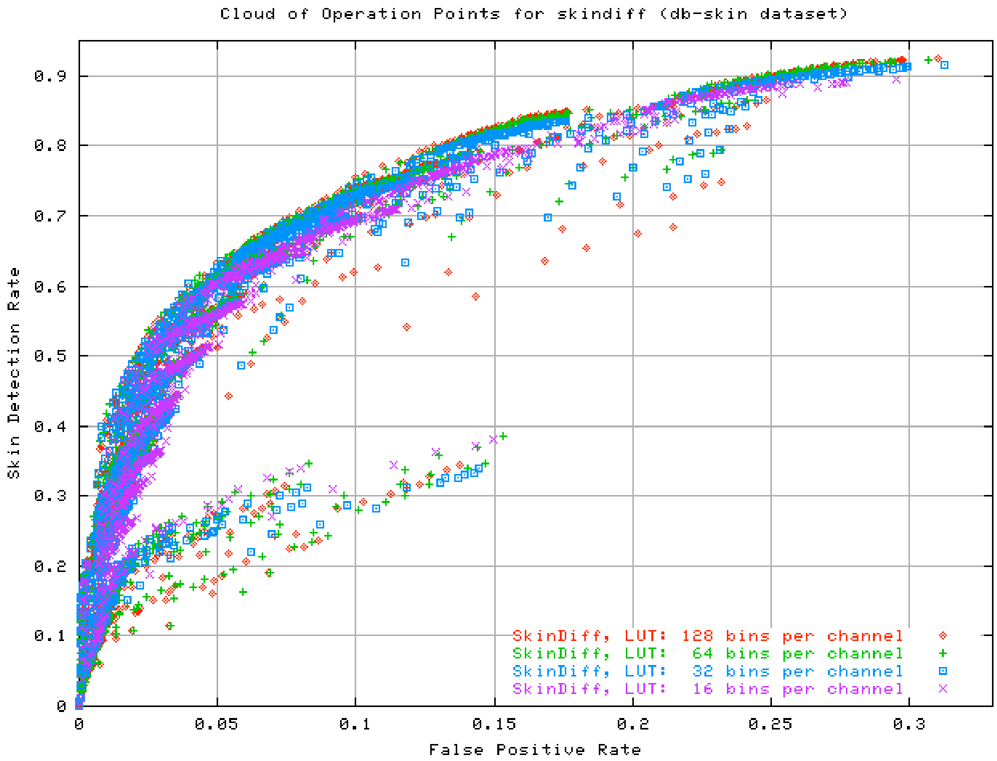


Figure 4: Cloud of the operation points for skindiff-j1 in the db-skin dataset. Different combinations of parameters' values were considered.

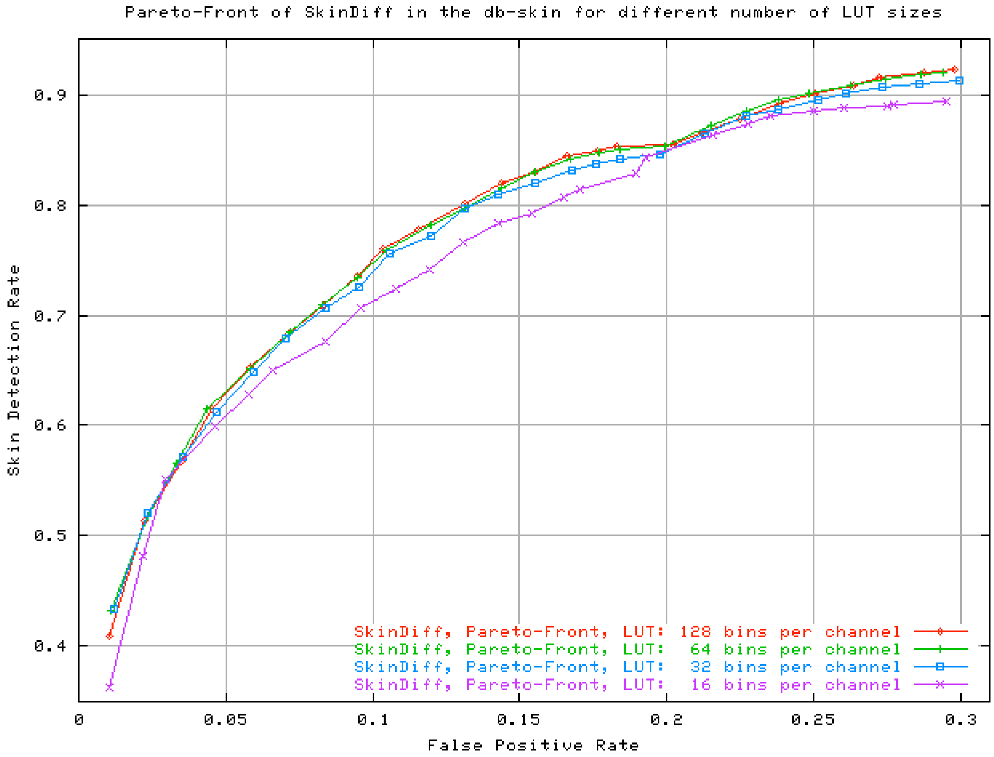
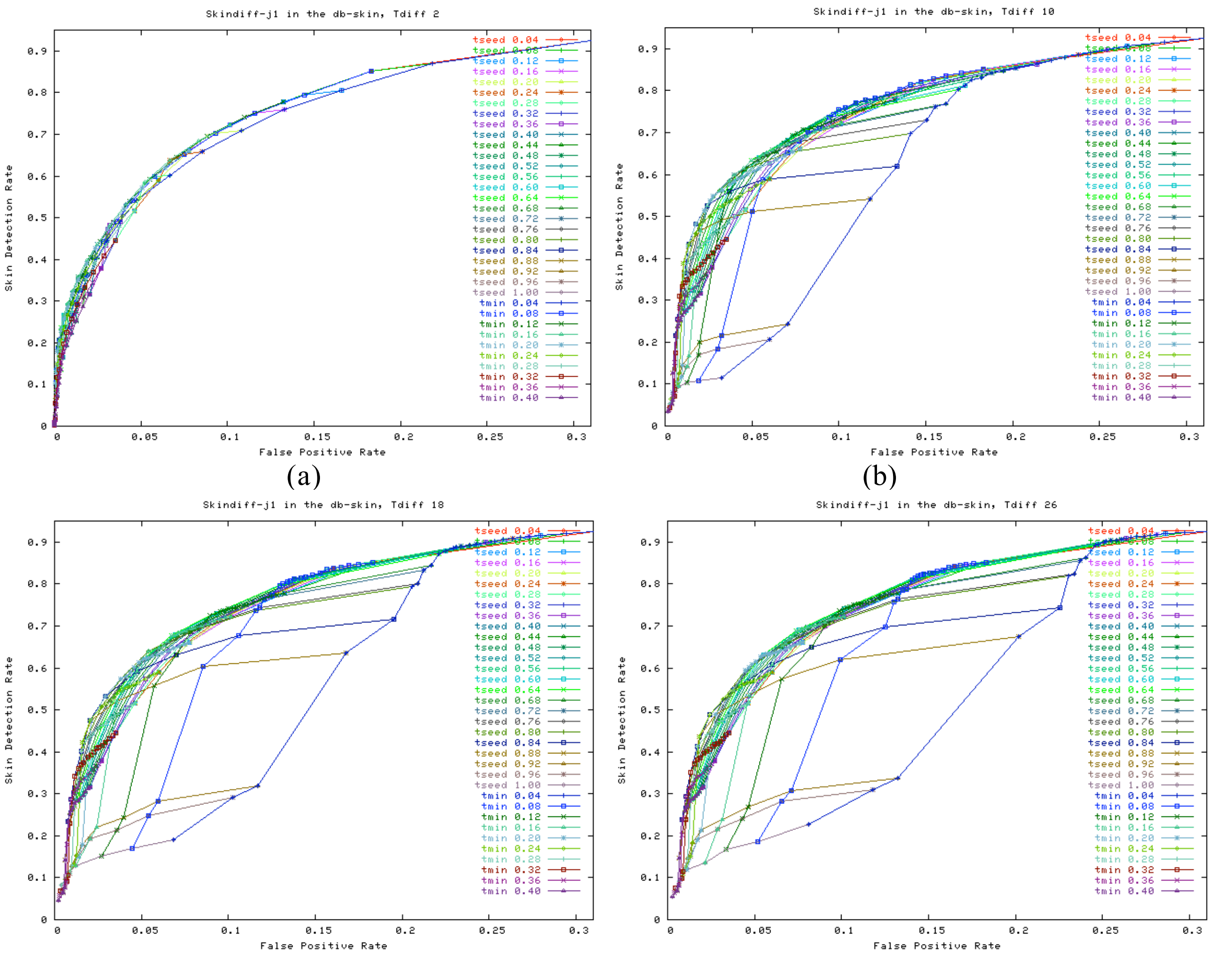


Figure 5. Best obtained ROC curves (Pareto-Front) for skindiff-j1 in the db-skin dataset.

After studying the LUT quantization effect in the algorithm, we analyze the behavior of the skindiff's parameters. This is not a trivial task because there are three parameters, and each one should be analyzed while keeping the other two fixed. For implementing this analysis, several sets of ROC curves using different combination of parameters were calculated. Each set of curves is parameterized by a single parameter. The curves corresponding to each set are generated by selecting different values for a second parameter, which is then kept fixed, and by moving the third parameter for building the corresponding curve. As an example, the set of ROC curves displayed in figure 6 (a) were built using T_{diff}=2. In the right part of the graph it can be observed the second parameter employed for building each curve, for example T_{seed}=0.08. In this case the corresponding curve (T_{diff}=2; T_{seed}=0.08) was built by moving the value of T_{min}.

By analyzing and comparing the set of curves displayed in figures 6 (a)-(d), the effect of T_{diff} in the performance of the algorithm can be observed. T_{diff} is the parameter that controls the smoothness of the skin regions, and together with T_{min} the extension of the diffusion process. As it can be seen in the graphs, for large values of T_{diff} and small values of T_{min} the performance of the algorithm is very poor, since too large and non-homogeneous skin regions are obtained. T_{diff} produces mainly "lateral" translations of the ROC curves, i.e. when T_{diff} is increased the FPR also increases but the DR does not change much. It can also be noticed that while keeping T_{diff} fixed, an increase of T_{seed} produces a reduction of the DR (less seeds are generated and some skin regions are not even considered in the diffusion process).

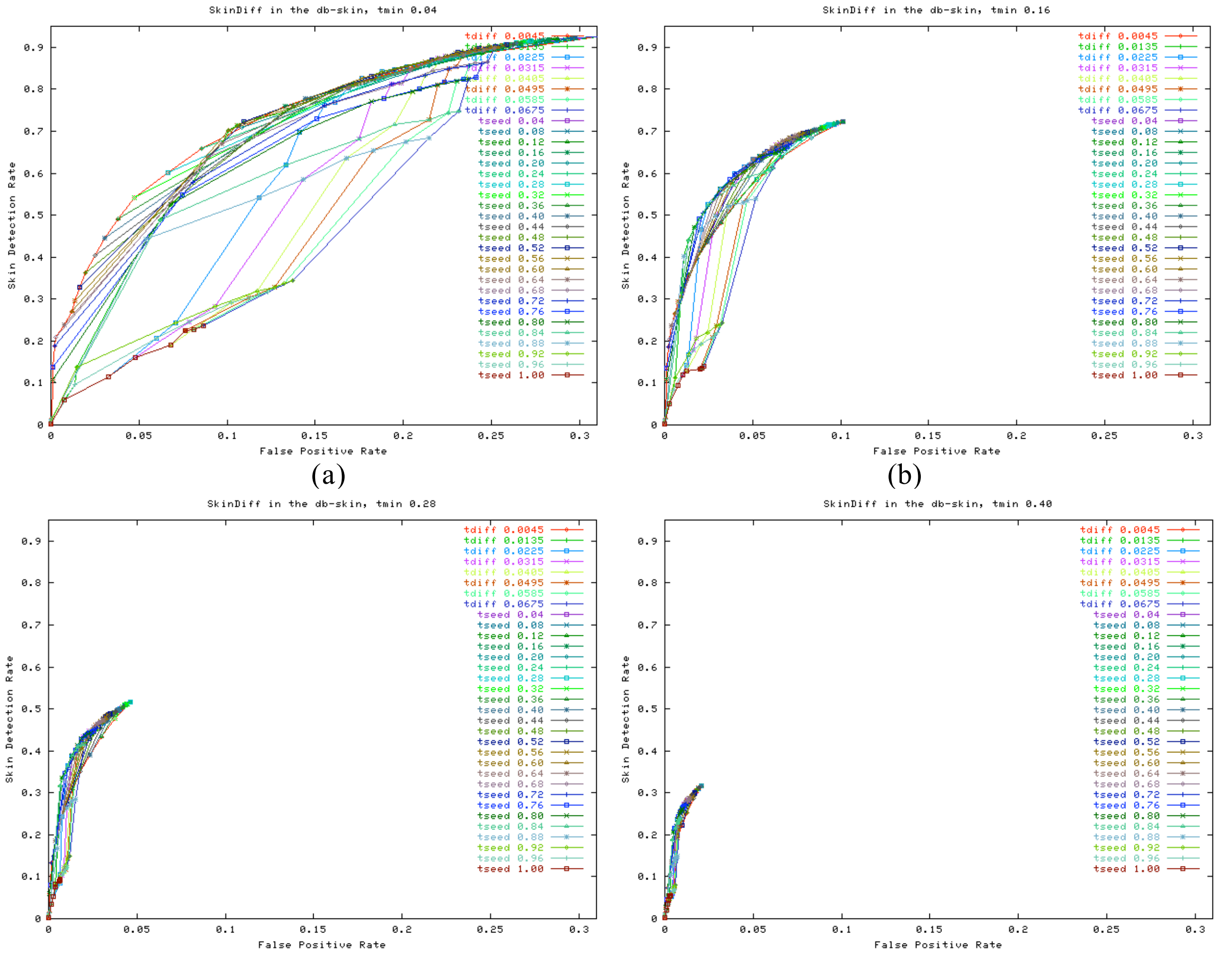


(a) (b)

(c) (d)

Figure 6. ROC curves for different values of T_{diff}: 2 (a), 10 (b), 18 (c) and 26 (d), using the db-skin dataset and 128 bins in each channel. T_{min} and T_{seed} are used as second parameter in the generation of the ROC curves.

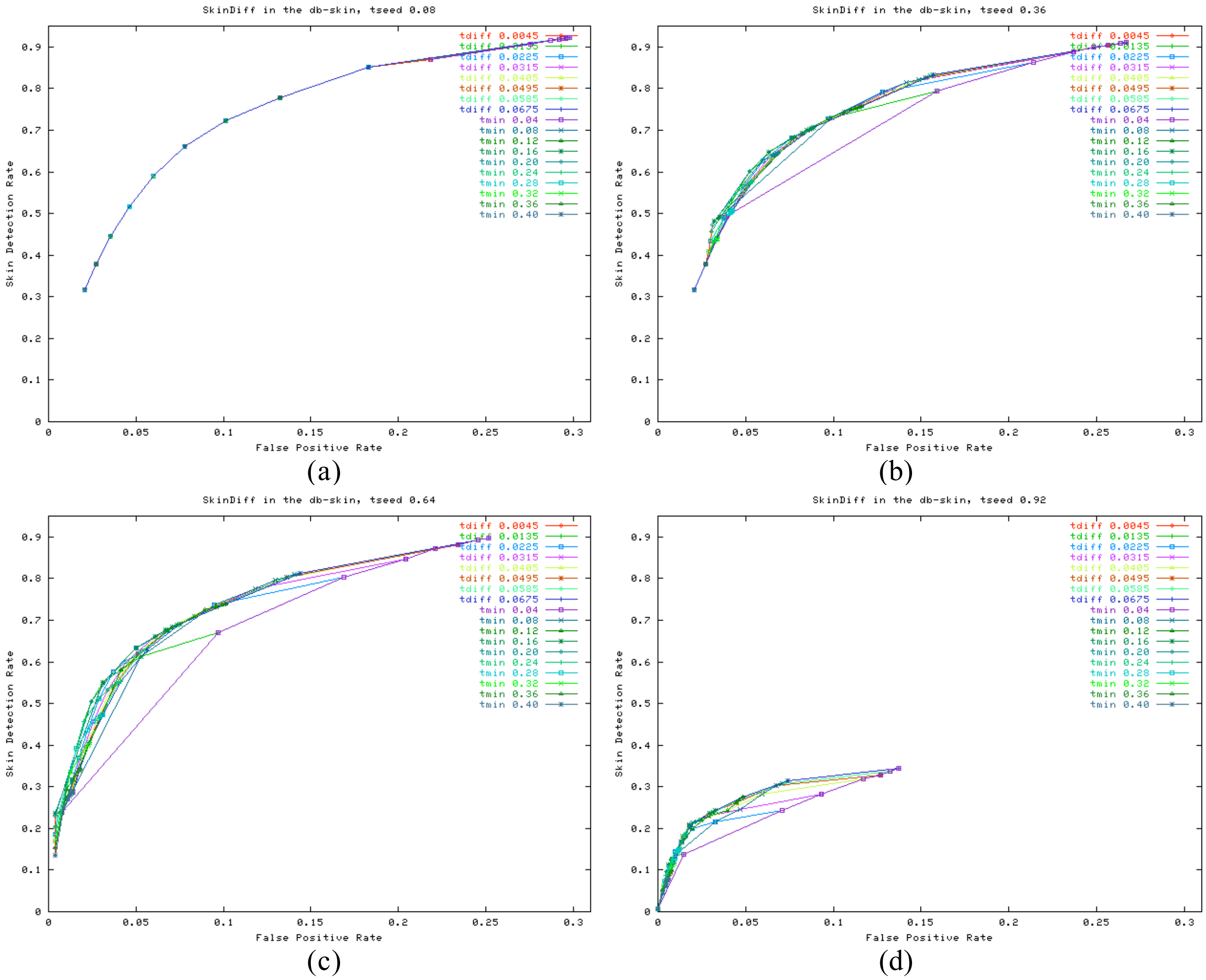
T_{min} defines the minimal acceptable skinness of a pixel and therefore the extension of the diffusion process. A large value of T_{min} produces a low DR, that because this large value generates a too restrictive criterion for selecting possible skin pixels (see the corresponding set of ROC curves in figure 7). Finally, T_{seed} defines the criterion for the generation of seeds pixels (see curves in figure 8). For large values of T_{seed} the performance of the algorithm is low. This happens because if T_{seed} is too large, few seeds are produced and large skin areas of the image are lost.



(a) (b)

(c) (d)

Figure 7. ROC curves for different values of T_{min}: 0.04 (a), 0.16 (b), 0.28 (c) and 0.40 (d), using the db-skin dataset and 128 bins in each channel. T_{seed} and T_{diff} are used as second parameter in the generation of the ROC curves.



(a) (b)

(c) (d)

Figure 8. ROC curves for different values of T_{seed}: 0.08 (a), 0.36 (b), 0.64 (c) and 0.92 (d), using the db-skin dataset and 128 bins in each channel. T_{min} and T_{diff} are used as second parameter in the generation of the ROC curves.

C. Skin Segmentation Evaluation

For testing the performance of skindiff, three baseline algorithms are employed Jones1, Jones2 and HSU. Jones1 corresponds to the skin classifier proposed by Jones&Rehg, using a MoG skin color model and a fixed decision threshold (T_{fix1}). Jones2 corresponds also to the skin classifier proposed by Jones&Rehg, but using the normalized ratio of a MoG skin color model and a MoG non-skin color model, and a fixed decision threshold (T_{fix2}). Jones1 and Jones2 were implemented using the same parameters employed in the original work [12]. As already mentioned, the parameters of the Jones&Rehg's skin models were also used for the MoG-based pixel-wise classifiers of skindiff-j1 (same as Jones1) and skindiff-j2 (same as Jones2). Thus, skindiff-j1 and skindiff-j2 correspond to extended versions of Jones1 and Jones2. HSU corresponds to the skin detection algorithm proposed in [10] (YCbCr, elliptical cluster model), but without using whitening compensation, for fairness in the comparison with the other algorithms. For getting different operation points for HSU, the ratios of the ellipse where increased and decreased, maintaining the proportionality between them. This ratio factor is called R_{scale}. In all methods, no post-processing (e.g. opening / closing) was employed.

The algorithms' evaluation was carried out using the db-tdsd-2, db-skin and db-nskin datasets. Before that, the parameters of skindiff-j1 and skindiff-j2 were tuned using the db-tdsd-1 dataset. The procedure was similar to the one employed for the parameter analysis (see former section). First, operation points corresponding to different combinations of the algorithms' parameters were found. Second, the Pareto-Front set (best parameter combinations) was determined. The selected parameters and their corresponding DR and FPR are listed in Table I.

Table I. Pareto-Front parameters for a skindiff-j1 and skindiff-j2, using the *db-tdsd-1* dataset. DR: Detection Rate. FPR: False Positive Rate.

Skindiff-j1 Pareto Front					**Skindiff-j2 Pareto Front**				
DR	**FPR**	T_{seed}	T_{min}	T_{diff}	**DR**	**FPR**	T_{seed}	T_{min}	T_{diff}
0.085063	0.00431618	0.94	0.20	6	0.065422	0.0092257	10.01	0.02	6
0.137303	0.0100298	0.90	0.27	18	0.168978	0.0130943	6.47	0.49	6
0.454245	0.0170793	0.78	0.24	10	0.302949	0.0169743	4.85	0.37	6
0.544195	0.0321571	0.71	0.12	6	0.413428	0.0243976	3.29	0.30	6
0.574978	0.0363111	0.63	0.20	10	0.525327	0.047495	3.27	0.27	10
0.615373	0.0434641	0.67	0.16	10	0.529743	0.0518615	4.79	0.30	18
0.654704	0.0613203	0.67	0.12	10	0.572621	0.063332	4.21	0.21	14
0.707378	0.0866388	0.67	0.12	18	0.578624	0.0691005	4.71	0.22	22
0.72035	0.095484	0.67	0.12	26	0.602341	0.0766728	5.17	0.18	22
0.779322	0.121245	0.59	0.078	14	0.634062	0.0952818	4.65	0.15	22
0.79358	0.129281	0.63	0.078	18	0.677061	0.113998	4.11	0.12	22
0.821379	0.149994	0.51	0.078	30	0.665285	0.117294	5.11	0.12	30
0.832122	0.158147	0.31	0.078	26	0.685272	0.133042	5.09	0.10	30
0.84543	0.175805	0.12	0.078	22	0.712982	0.14916	3.59	0.09	30
0.897492	0.24477	0.43	0.04	22	0.757563	0.162411	3.07	0.08	30
0.903595	0.254578	0.51	0.04	30	0.807593	0.202336	3.05	0.05	30

In figure 9 are shown the ROC curves, using the *db-tdsd-2* dataset, of the 5 compared algorithms. By analyzing the ROC curves it can be noticed that the best performing algorithms are Jones2, skindiff-j1 and skindiff-j2. After them Jones1 comes, and finally HSU. In the upper regions of the ROC curves, the best performing algorithm is Jones 2, while in the lower regions best performance is achieved by skindiff-j1 and skindiff-j2. It is important to notice that the lower regions of the curve correspond to low values of FPR. Normally, image regions that are difficult to segment (changing lighting conditions and background with skin-like colors) have, for a given FPR, a lower DR compared to non-difficult image regions. Therefore, we hypothesize that lower regions of the ROC curves correspond mainly to difficult to segment image areas, and that these areas are better segmented by skindiff-j1 and skindiff-j2. This hypothesis is confirmed when processing the images of the db-skin dataset, which contains images that are very difficult to segment (see [34]). As it can be observed in figure 10, skindiff-j1 outperforms all algorithms in the db-skin dataset. Interesting to see is that although without using a non-skin color model, skindiff-j1 outperforms skindiff-j2 and Jones1 outperforms Jones2. The reason seems to be that the MoG computed in [12] does not model accurately the non-skin pixels' probability distribution.

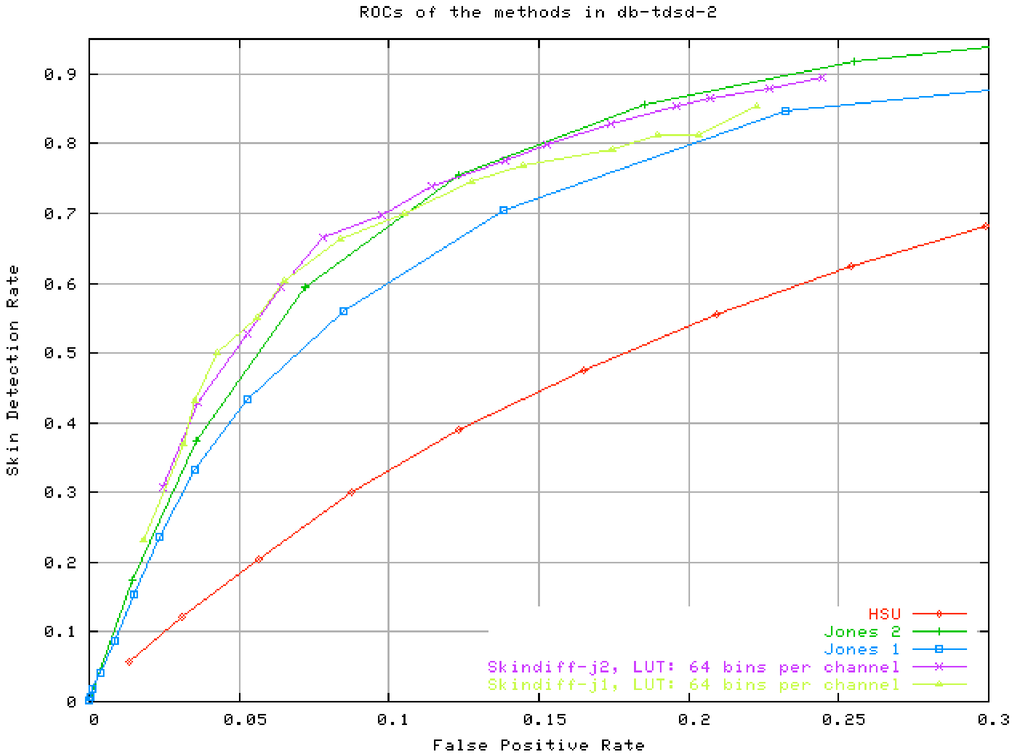


Figure 9. ROC Curves of the compared algorithms in the db-tdsd-2 database (see main text for details).

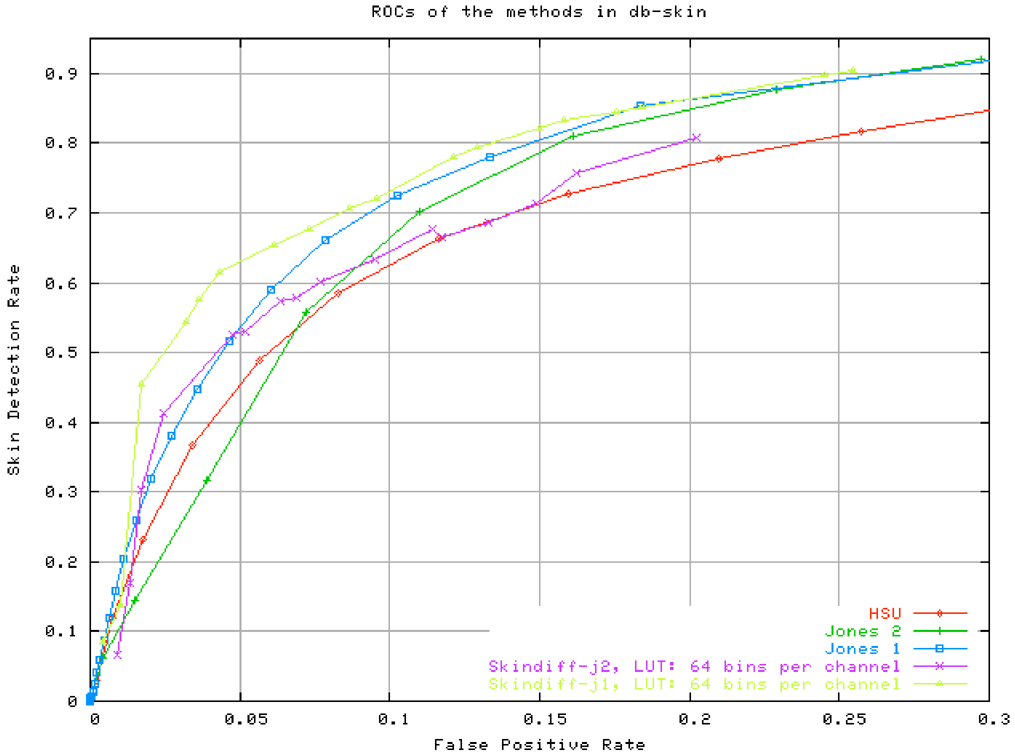


Figure 10. ROC Curves of the compared algorithms in the db-skin database (see main text for details).

Table II: Parameters and results in terms of detection rate (DR) and false positive rate (FPR) for the db-skin dataset. The results are organized in three groups that correspond to three different DR values: 0.62, 0.72 and 0.82. This table also includes information of the false positive rate of the compared algorithms for the db-nskin dataset (FPR2).

Method	**Parameters**			**Exact DR**[2]	**FPR**	**FPR2**
Group 1	**DR:0.62**					
HSU	R_{scale}: 0.65	-	-	0.6261	0.0976	0.0788
Jones 1	T_{fix1}: 0.22	-	-	0.6213	0.0649	0.0563
Jones 2	T_{fix2}: 3.85	-	-	0.6290	0.0864	0.0822
skindiff-j1	T_{seed}: 0.67	T_{min}: 0.16	T_{diff}: 10	0.6153	**0.0434**	**0.0253**
skindiff-j2	T_{seed}: 4.65	T_{min}: 0.15	T_{diff}: 22	0.6340	0.0950	0.0655
Group 2	**DR:0.72**					
HSU	R_{scale}: 0.8	-	-	0.7278	0.1596	0.1172
Jones 1	T_{fix1}: 0.16	-	-	0.7249	0.1022	0.0855
Jones 2	T_{fix2}: 2.30	-	-	0.7207	0.1174	0.1084
skindiff-j1	T_{seed}: 0.67	T_{min}: 0.12	T_{diff}: 26	0.7203	**0.0954**	**0.0602**
skindiff-j2	T_{seed}: 3.59	T_{min}: 0.09	T_{diff}: 30	0.7129	0.1491	0.0995
Group 3	**DR:0.82** (Reference point form the original HSU algorithm [10])					
HSU	R_{scale}: 1.0	-	-	0.8164	0.2571	0.1800
Jones 1	T_{fix1}: 0.1	-	-	0.8144	**0.1544**	0.1284
Jones 2	T_{fix2}: 1.28	-	-	0.8109	0.1612	0.1413
skindiff-j1	T_{seed}: 0.31	T_{min}: 0.08	T_{diff}: 26	0.8213	0.1581	**0.1222**
skindiff-j2	T_{seed}: 3.05	T_{min}: 0.05	T_{diff}: 20	0.8075	0.2023	0.1291

For verifying that the algorithms' parameters are not over-fitted to the db-tdsd-2 and db-skin datasets, we perform experiments with the db-nskin dataset (5,882 images, 1,567 million non-skin pixels). Using the same parameters employed for obtaining three reference points corresponding to a DR of 0.62, 0.72 and 0.82 in the db-skin dataset, we performed skin detection in the db-nskin dataset using all compared algorithms. The obtained results in terms of FPR are displayed in table II (right most column, FPR2). As it can be observed the best results (lowest FPR) are obtained by skindiff-j1, followed by Jones1, skindiff-j2, Jones2 and HSU (in this order). We notice again that skindiff-j1 outperforms skindiff-j2, and that Jones1 outperforms Jones2. The reason seems to be the inaccurate MoG non-skin model. For analyzing this situation from another point of view, the FPRs corresponding to the db-skin dataset were displayed against the FPRs corresponding to the db-nskin dataset, for all algorithms under analysis. In figure 11 are shown curves for each algorithm. The curve reference points Opt1, Opt2 and Opt3, correspond to DR=0.82, 0.72 and 0.62, respectively. The best performing algorithms are the ones with reference points closer to the origin (lower FPR in both axis). It can be seen that in all three cases, the best performing algorithm is skindiff-j1, followed by Jones1.

[2] ROC curves contain a discrete number of values. For this reason real DR values approximate the 3 given reference points (0.62, 0.72, 0.82).

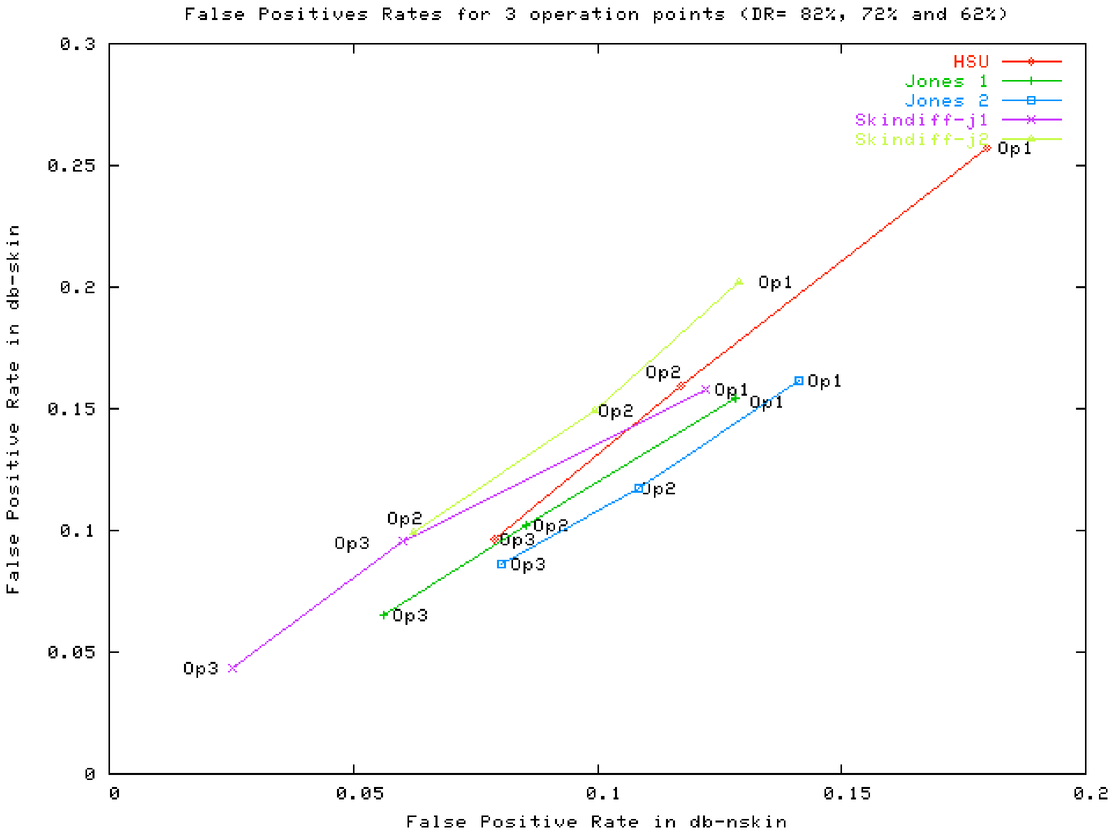


Figure 11. FPR in the db-skin dataset against FPR in the db-nskin dataset. Op1: DR=0.82, Op2: DR=0.72, and Op3: DR=0.82 (DR value calculated in the db-skin dataset).

After analyzing many skin segmentations results we conclude that, when processing real world images the skindiff algorithms main strengths are: (i) good performance when detecting skin in bright image areas and in areas with illumination producing specular lighting (highlights); (ii) adequate processing of backgrounds since skin seeds seldom appear in background areas; and (iii) less required post-processing of the skin detections; it is not necessary to filter isolated skin pixels and the filling of holes can be done using a smaller dilation filter (fast application). These characteristics will be analyzed using three exemplary images. These images are processed by all compared methods, using an operation point of DR=72% in the db-skin dataset.

In figure 12 is shown an image containing a woman with frontal illumination. As it can be observed, highlights are present in the face region, especially in the nose and forehead, as well as in the arms and hands. Skin segmentations results of all compared methods, as well as skin seeds pixels corresponding to skindiff-j1 and skindiff-j2, are shown. As it can be observed best segmentation results are obtained by skindiff-j2. The algorithm segments correctly the bright body areas, including the highlights of the nose, forehead, arms and hands. Only small holes (non-segmented pixels) are left in some face's and arms' areas. These holes could be easily filled using post-processing. It can also be seen that skindiff-j2 does not detect any skin in the background or in the lady's hair, as other algorithms do. When observing the corresponding skin seeds image, it can be noticed that the reason of this good behavior is the occurrence of the skin seeds only in skin areas (face and arms). When analyzing the other algorithms it can be seen that HSU, Jones1 and Jones2 cannot deal correctly with the bright skin areas. Worse results are obtained by HSU. Jones1 also achieves bad segmentation results, which produces a bad behavior of skindiff-j1. In any case skindiff-j1 outperforms Jones1, as skindiff-j2 outperforms Jones2.

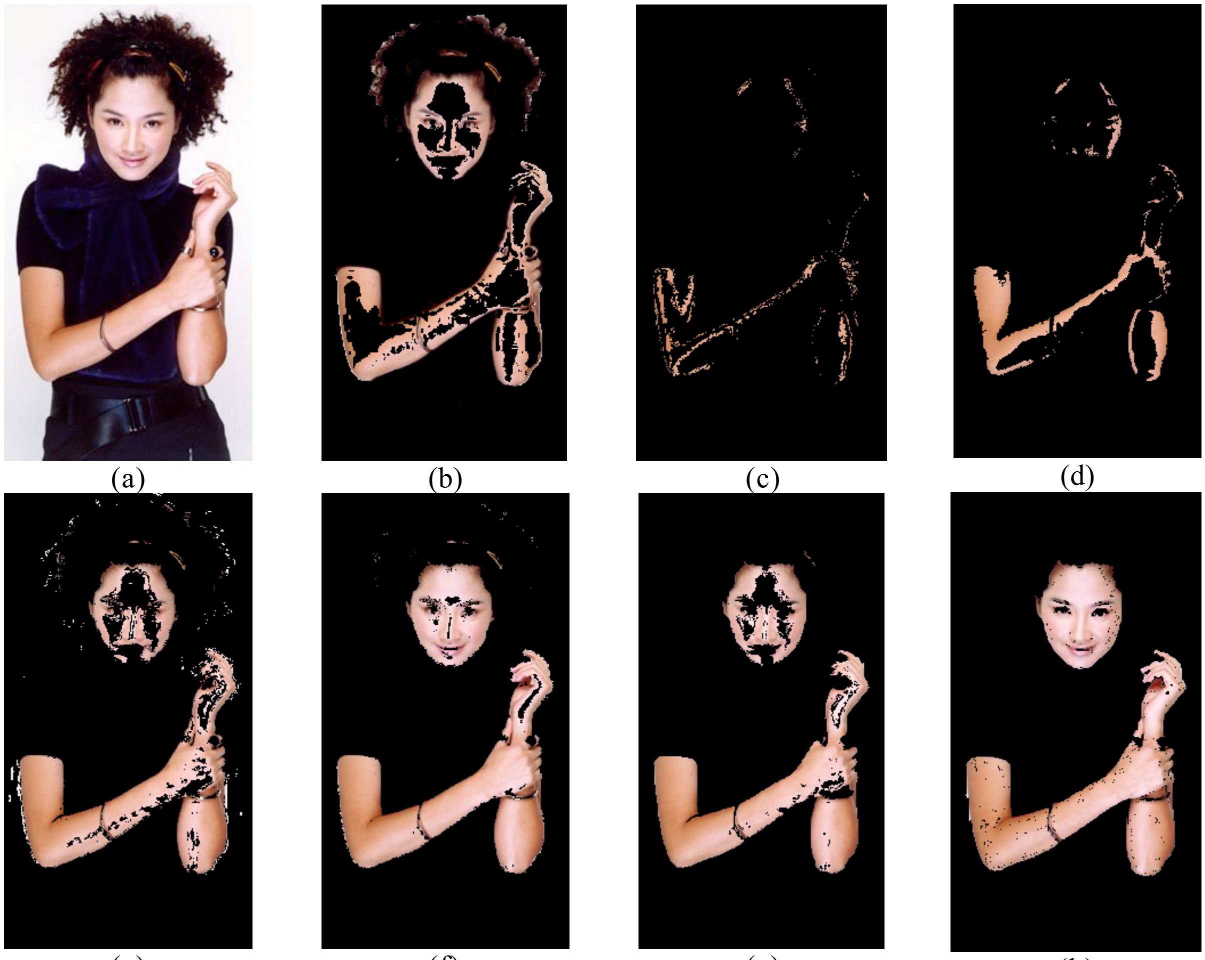


Figure 12. Example 1. Skin segmentation of an image with specular lighting. Original image (a). Skin segmentation results using HSU (b), Jones1 (e), Jones2 (f), skindiff-j1 (g) and skindiff-j2 (h). Seed's pixels corresponding to skindiff-j1 (c) and skindiff-j2 (d).

In figure13 is shown an image containing three women with skin-like colors in the background. As it can be noticed, highlights are also present in some body region (legs, arms, forehead, etc.). Skin segmentations results of all compared methods, as well as skin seeds pixels corresponding to skindiff-j1 and skindiff-j2, are shown. As it can be observed best segmentation results are obtained by skindiff-j1 and skindiff-j2. Skindiff-j1 outperforms Jones1, while skindiff-j2 outperforms Jones2. Worse results are obtained by HSU. Skindiff-j1 segments correctly the body areas, including the highlights, but it makes some mistakes when segmenting, as skin, a skin-like wall (right part of the image), and some parts of the floor. Skindiff-j2 segments almost correctly the body areas but it makes some mistakes in the highlights regions. Moreover, when processing the background it segments wrongly, as skin, some parts of the bikinis, a region of the floor and some small regions of the building behind the ladies.

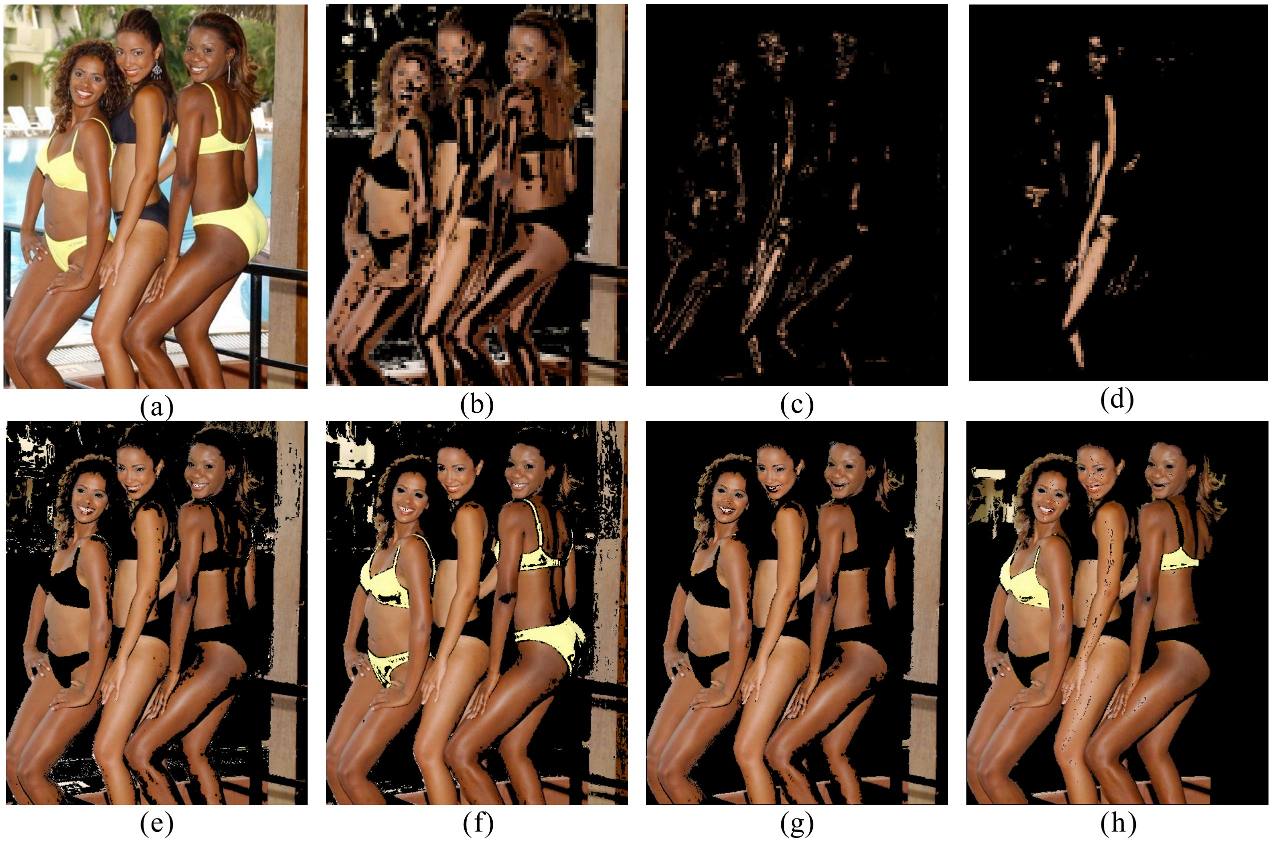
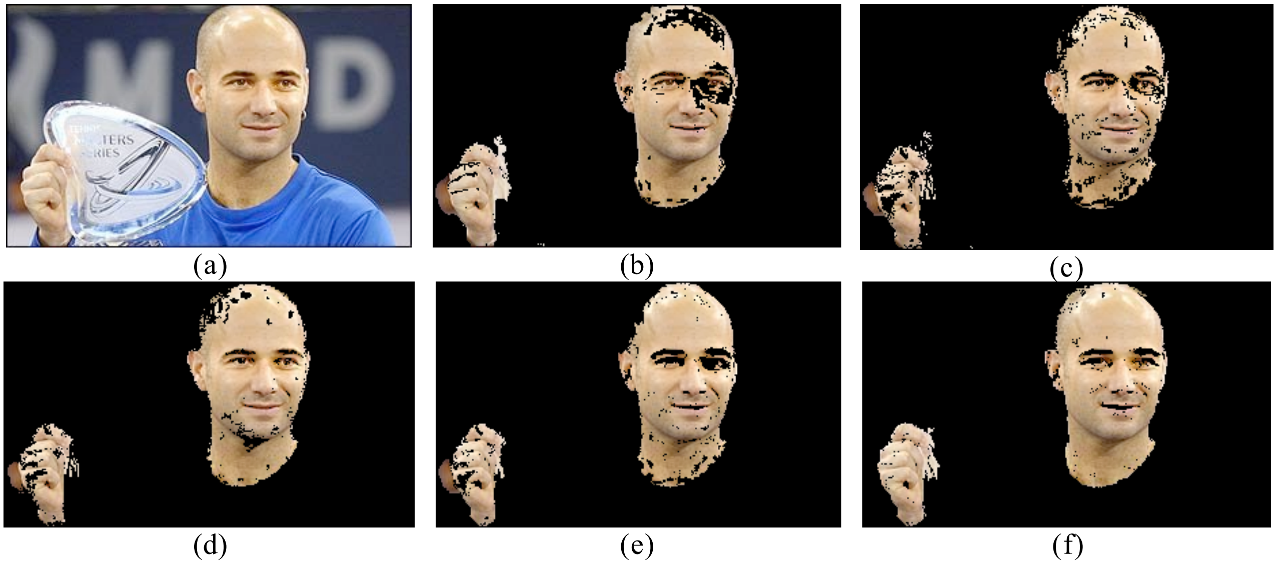


Figure 13. Example 2. Skin segmentation of an image with specular lighting and background with skin-like colors. Original image (a). Skin segmentation results using HSU (b), Jones1 (e), Jones2 (f), skindiff-j1 (g) and skindiff-j2 (h). Seed's pixels corresponding to skindiff-j1 (c) and skindiff-j2 (d).

In figure 14 is shown an image of Andre Agassi, having strong highlights in the head area. Skin segmentations results of all compared methods are shown. As it can be observed best segmentation results are obtained by skindiff-j2, followed by skindiff-j1. As in the former examples, skindiff-j2 outperforms Jones2, skindiff-j1 outperforms Jones1, and the worst results are obtained by HSU. Skindiff-j2 segments correctly the strong highlights of the face area, and only small holes are left. These holes could be easily filled using post-processing.



(a) (b) (c)

(d) (e) (f)

Figure 14. Example 3. Skin segmentation of an image with specular lighting and shadows. Original image (a). Skin segmentation results using Hsu (b), Jones1 (c), Jones2 (d), skindiff-j1 (e) and skindiff-j2 (f).

The proposed skindiff-j1 and skindiff-j2 are not only robust when detecting skin, but also fast. In table III is shown the processing time when detecting skin in the image of 217x400 pixels (23% skin) shown in figure 12. As it can be seen, skindiff-j1 and skindiff-j2 process this image in about 100ms, outperforming the HSU algorithm, and the original implementations of Jones1 and Jones2, which are more than 10 times slower. Fast implementations of Jones1 and Jones2 (LUT-64) are about 10% faster than skindiff-j1 and skindiff-j2. However, considering robustness (DR + FPR) and processing time, we believe that the best performing algorithms are skindiff-j1 and skindiff-j2.

Table III. Processing time of the image shown in figure 12 (217x400 pixels; 23% skin). The processing conditions are: Pentium 4 1.8Ghz, 512 RAM, Linux 2.4 (Red Hat 3.2.3).

Algorithm	**Time [s]**
skindiff-j1, LUT-64	0.0954
skindiff-j2, LUT-64	0.1038
HSU	0.1361
Jones1, LUT-64	0.0899
Jones2, LUT-64	0.0899
Jones1, no LUT (original)	1.2381
Jones2, no LUT (original)	2.1980

V. Conclusions

This work is based on the assumption that the pixel-wise skin detection paradigm, in which each image pixel is individually analyzed, should be extended when dealing with real-world conditions (changing lighting conditions and complex backgrounds containing surfaces and objects with skin-like colors), and context information needs to be incorporated in the skin detection process. On base of this idea, in this report was proposed a robust skin segmentation approach that uses local spatial context, i.e. the decision about the class (skin or non-skin) of a given pixel considers information about the pixel's neighbors. This approach is defined by two main stages: pixel-wise classification and spatial diffusion. The pixel-wise classification stage can be implemented using any standard skin detection algorithm applied over a given color space. Afterwards, the spatial diffusion stage will refine the pixel-wise skin detection results, by using neighborhood information for determining the final skin regions. Therefore, using our approach different skin segmentation algorithms can be derived.

The proposed approach is efficiently implemented by the skindiff algorithm, which achieves a robust segmentation of skin regions at a high processing speed. Skindiff uses the RGB color space, MoG models implemented using LUTs as pixel-wise classification algorithm, and an efficient implementation of the spatial diffusion. Two flavors of skindiff were analyzed in this report, skindiff-j1 that uses a single MoG skin color model, and skindiff-j2 that in addition to a MoG skin color model employs a MoG non-skin color model. Experimental results of skindiff-j1 and skindiff-j2 were presented using three datasets composed by real-world images (db-tdsd-2, db-skin, db-nskin). Skindiff-j1 and skindiff-j2 were favorably compared with three baseline algorithms: they outperform all three algorithms in terms of DR and FPR, by having a comparable processing time.

The algorithms' dependence on its parameters was also analyzed in this report. Some positive features of skindiff-j1 and skindiff-j2, observed during our experimental analysis, are: (i) correct detection of skin in bright image areas and in areas with illumination producing specular lighting; (ii) adequate processing of backgrounds since skin seeds seldom appear in background areas; and (iii) less required post-processing of the skin detections since it is not necessary to filter isolated skin pixels and the filling of holes can be done using a smaller dilation filter (fast application). It should be mentioned that although most of the time skindiff-j2 performs slightly better than skindiff-j1, when complex backgrounds having pixels with skin-like colors are considered, skindiff-j1 outperforms skindiff-j2. The reason seems to be that the employed MoG cannot correctly model the non-skin pixels' distribution. Therefore, one future improvement of skindiff-j2 could be a better modeling of the non-skin pixels' distribution.

Another issue that needs to be investigated for a further improving of the algorithm's performance is the correct generation of seeds in all skin areas. As a future work we also want to extend our approach for the processing of videos by including temporal context in the skin segmentation process. We are also thinking on including an adaptable skin model that can be calculated at high processing speed. Finally, we are also interested on having adaptable algorithms for selecting the diffusion parameters (e.g. genetic algorithms).

References

[1] M. Abdel-Mottaleb, and A. Elgammal, "Face detection in complex environments from color images", *Proc. IEEE Int. Conf. on Image Processing*, Kobe, Japan, 3: 622-626, 1999.

[2] A. Albiol, L. Torres, Ed. Delp, "An Unsupervised Color Image Segmentation Algorithm for Face Detection Applications", *Proc. IEEE Int. Conf. on Image Proc. – ICIP 2001*, Greece, 2001.

[3] A. Albiol, L. Torres, Ed. Delp, "Optimum Color Spaces for Skin Detection", *IEEE Int. Conf. on Image Proc. – ICIP 2001*, Greece, 2001.

[4] D. Chai, and K.N. Ngan, "Locating facial region of a head-and-shoulders color image" *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, 124-129, 1998.

[5] Q. Chen, H. Wu, and M. Yachida, "Face detection by fuzzy pattern matching", *Proc. 5th Int. Conf. on Computer Vision*, Cambridge, Massachusetts, USA, 591-596, 1995.

[6] T.Y. Chow, and K.M. Lam, "Mean-shift based mixture model for face detection in color image", *Proc. IEEE Int. Conf. on Image Proc. – ICIP 2004*, 601 – 604, Vol. 1, 24 – 27 Oct. 2004.

[7] C. Coello, D. Van Veldhuizen, G. Lamont, "Evolutionary Algorithms for Solving Multi-objective Problems", Kluwer Academic Publishers 11-13, 2002.

[8] Y. Dai, and Y. Nakano, "Extraction for facial images from complex background using color information and SGLD matrices", *Proc. 1st Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, 238-242, 1995.

[9] J. Fan, D.K.Y. Yau, A.K. Elmagarmid, and W.G. Aref, “Automatic image segmentation by integrating color-edge extraction and seeded region growing”, *IEEE Trans. on Image Processing*, Volume 10, No. 10, 1454 – 1466, Oct. 2001.

[10] R. L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, “Face detection in color images”, *IEEE Trans. on Pattern Anal. and Machine Intell.* 24(5): 696-706, 2002.

[11] B. Jedynak, H. Zheng, and M. Daoudi, “Statistical Models for Skin Detection”, *IEEE Workshop Statistical Analysis in Computer Vision*, together with CVPR 2003, June 22, 2003.

[12] M.J. Jones, and J.M. Rehg, “Statistical color models with application to skin detection”, *Int. Journal of Computer Vision* 46(1): 81-96, 2002.

[13] S. Karungaru, M. Fukumi, and N. Akamatsu, “Feature Extraction for Face Detection and Recognition”, *Proc. 13th IEEE Int. Workshop on Robot and Human Interactive Communication*, RO-MAN 2004, 235 – 239, 20-22 Sept. 2004.

[14] S. Kawato, and J. Ohya, “Real-time detection of nodding and head-shaking by directly detecting and tracking the between eyes”, *Proc. 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, 40-45, 2000.

[15] D. Kottow, M. Köppen, and J. Ruiz-del-Solar, “A Background Maintenance Model in the Spatial-Range Domain”, *2nd Workshop on Statistical Methods in Video* Processing (ECCV 2004 associated workshop), Prague, Czech Republic, May 16, 2004.

[16] H. Kruppa, and B. Schiele, “Using Local Context To Improve Face Detection”, *Proc. British Machine Vision Conf. – BMVC 2003*, Norwich, UK, Sept. 2003.

[17] C.-H. Lin and J.-L. Wu, “Automatic facial feature extraction by genetic algorithms”, *IEEE Trans. on Image Processing*, Vol. 8, No. 6, 834 – 845, June 1999.

[18] B. Martinkauppi, *Face Color under Varying Illumination – Analysis and Applications*, Doctoral Thesis, University of Oulu, Finland, 2002.

[19] E. Saber, and A.M. Tekalp, “Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions”, *Pattern Recognition Letters* 17(8): 669-680, 1998.

[20] S. Satoh, Y. Nakamura, and T. Kanade, “Name-it: naming and detecting faces in news videos”, IEEE Multimedia 6(1): 22-35, 1999.

[21] D. Saxe, and R. Foulds, “Toward robust skin identification in video images”, *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, Killington, Vermont, USA, 379-384, 1996.

[22] M. Shin, K. Chang, and L. Tsap, “Does colorspace transformation make any difference on skin detection?”, *Proc. IEEE Workshop on Appl. of Computer Vision 2002*, Florida, USA, 2002.

[23] L. Sigal, S. Sclaroff, and V. Athisos, “Skin color-based video segmentation under time-varying illumination”, *IEEE Trans. on Pattern Anal. and Machine Intell.* 26(7): 862-877, 2004.

[24] Q. B. Sun, W.M. Huang, and J. K. Wu, “Face detection based on color and local symmetry information”, *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, 130-135, 1998.

[25] J. Ruiz-del-Solar and R. Verschae, “Skin Detection using Neighborhood Information”, *Proc. 6th Int. Conf. on Face and Gesture Recognition – FG 2004*, 463 – 468, Seoul, May 17 – 19, 2004.

[26] J. Ruiz-del-Solar and R. Verschae, “Robust Skin Segmentation using Neighborhood Information”, *Proc. Int. Conf. on Image Processing - ICIP 2004*, 207 – 210, Singapore, Oct. 24 – 27, 2004.

[27] N. Sebe, I. Cohen, T. Huang, T. Gevers, “Skin Detection: A Bayesian Network Approach”, *Proc. 17th Int. Conf. on Pattern Recognition – ICPR 2004*, Vol. 2, 903 – 906, Cambridge, UK, Aug. 23 – 26, 2004.

[28] L. Spillman and J. Werner (Eds.), *Visual Perception: The Neurophysiological Foundations*, Academic Press, 1990.

[29] M. H. Yang, and N. Ahuja, “Detecting human faces in color images”, *Proc. IEEE Int. Conf. on Image Processing*, Chicago, Illinois, USA, 1: 127-130, 1998.

[30] L. Yin, J. Jia, and J. Morrissey, “Towards Race-Related Face Identification: Research on Skin Color Transfer”, *Proc. 6th Int. Conf. on Face and Gesture Recognition – FG 2004*, 362 - 368, Seoul, May 17 – 19, 2004.

[31] H. Zheng, M. Daoudi, and B. Jedynak, “From Maximum Entropy to Belief Propagation: An application to Skin Detection”, *Proc. British Machine Vision Conf. – BMVC 2004*, Kingston upon Thames, London, UK, Sept. 7 – 9, 2004.

[32] Q. Zhu, K.-T. Cheng, C.-T. Wu, and Y.-L. Wu, “Adaptive Learning of an Accurate Skin-Color Model”, *Proc. 6th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 37-42, Seoul, Korea, May 17 – 19, 2004.

[33] Q. Zhu, C.-T. Wu, K.-T. Cheng, and Y.-L. Wu, “An Adaptive Skin Model and Its Application to Objectionable Image Filtering”, Pr*oc. 12th Annual ACM Int. Conf. on Multimedia*, 56 – 63, New York, USA, Oct. 10-16, 2004.

[34] http://vision.die.uchile.cl/skindiff