Skin Detection in Videos in the Spatial-Range Domain*

Javier Ruiz-del-Solar[1,2], Rodrigo Verschae[1,2], and Daniel Kottow[1]

[1] Department of Electrical Engineering, Universidad de Chile
[2] Center for Web Research, Department of Computer Science, Universidad de Chile
{jruizd, rverscha}@ing.uchile.cl

Abstract. Most of the already proposed skin detection approaches are based on the same pixel-wise 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. Following this idea, in this article is proposed a robust and fast skin detection approach that uses spatial and temporal context. Spatial context implies that the decision about the class (skin or non-skin) of a given pixel considers information about the pixel's neighbors. Temporal context implies that skin detection is carried out considering not only pixel values from the current frame, but also taking into account past frames and general background reference information.

1 Introduction

Skin detection is a very popular and useful technique for detecting and tracking human-body parts. 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. Changing lighting conditions and complex backgrounds containing surfaces and objects with skin-like colors are major problems 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, normalized RGB, HIS-HSV, YCbCr, YIQ, YES, YUV, CIE XYZ, CIE LUV, and Lab (see [11] for references). We believe that just selecting the "best" color space does not solve the mentioned drawbacks [5][1]. Some authors have used statistical models for solving the skin/non-skin classification problem (Mixture of Gaussians (MoG) [8][3] and histogram models [3][2]). In [9] a model that adapts a MoG to the image contents is used. In [6] an approach for skin detection under time-varying illumination using adaptive color histograms that model the color distribution over time is proposed. We think that statistical models are in the right direction; however they miss the benefits of using context information.

All mentioned approaches are based on the same pixel-wise paradigm, in which each image pixel is individually analyzed. We think that this paradigm should be

* This research was funded by Millenium Nucleus Center for Web Research, Grant P04-067-F, Chile.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 961 – 969, 2005.

extended incorporating context information to the skin detection process. Human *beings* can detect skin in real scenes, 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, because human skin detection is not a simple low-level process, but a process in which high-level mechanisms are also involved. If we think on the human perception of a blue ball under variable illumination, we will agree in 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. For having 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 [7]. In the same way, the detection of skin in a face or 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) [7].

Following these ideas we proposed a robust and fast skin detection approach that uses spatial context (spatial neighborhood information) and temporal context (temporal neighborhood and foreground pixels information). Spatial context implies that the decision about the class (skin/non-skin) of a given pixel considers information about the pixel's spatial neighbors. A diffusion process 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 color between neighbor pixels changes smoothly and at the same time the pixel *skinness*[1] keeps a minimal acceptable value. The seeds of the diffusion process are pixels with a high skinness value. Temporal context implies that skin detection is carried out considering the belonging of a pixel to the current background (BG) or foreground (FG) model. More explicitly, we model pixels belonging to the BG and pixels belonging to the FG by two sets of finite codebook vectors in the so-called spatial-range domain. In this domain pixels whose values are close enough to the BG or FG codebooks' vectors are classified as BG or FG, respectively. Given that we have codebooks "representing" (quantizing) the image pixels, we apply the spatial diffusion process only to the codebooks' vectors. These sets are much smaller than the number of image pixels in each frame and consequently we achieve a much faster processing time.

It is important to mention that the here-proposed work corresponds to an extension and integration of our algorithms: (i) *skindiff* [10], a skin detection algorithm for static images that uses local spatial context, and (ii) *tracey* [4], a BG and FG maintenance algorithm that works in the spatial-range domain.

2 The Proposed Skin Detection System

As already mentioned, the proposed skin detection systems works in the spatial range domain, incorporating spatial and temporal context information. Spatial context information for the skin detection is introduced using a diffusion process. Temporal

[1] We define Skinness as the belonging of a pixel to the skin class.

context is introduced by maintaining a BG/FG model, and by running the skin detection algorithm in: (1) the BG prototypes that were previously classified as skin, and (2) the FG prototypes. In this way the system is capable of dealing with static or slowly moving skin areas.

2.1 Vector Quantization of an Image on the Spatial-Range Domain

The spatial-range domain takes into account simultaneously the location and the intensity values of the image pixels. In this domain each image pixel has two parts, a spatial part (position) and a range part (intensity), where the range part may be written as a function of the spatial part [4]:

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

The superscripts *s* and *r* denote the spatial (location) and range (value, intensity) parts of the pixels, respectively. By using this domain space an image can be *represented* (quantized or encoded) by a set of codebook vectors $C = \{c_i\}_{i=1..m}$ that cover all the images pixels. Each codebook vector will represent an equally-sized portion of the spatial-range domain, namely, all vectors lying inside a constant-sized hyper rectangle[1] $\bar{c}_i$ centered at the codebook vector:

$$x_j \in \bar{c}_i \Leftrightarrow \left\| x^s - c_i^s \right\| < \sigma_s \wedge \left\| x^r - c_i^r \right\| < \sigma_r$$

The size of the rectangle is given by σ_s and σ_r, which are constant and independent of *i*. They define the accuracy of the representation in the spatial and range sub domains. We say that a complete image *I* is represented accurately by a set of vectors C in the spatial range-domain iff:

$$\forall x_j \in I \, \exists c_i \in C (x_j \in \bar{c}_i) \quad (2)$$

Thus, using the algorithm shown in Figure 1 a given image can be learned, i.e. represented by a set of codebook vectors. Figure 2 shows an example of representation using the Lena image.

2.2 Tracey: The Background Model

Our BG model represents the BG by a set of codebook vectors in the spatial-range domain; image pixels not represented by this set will be output as the segmented FG To cope with the intensity variations of the BG, this set of codevectors is continuously updated. Our BG maintenance model consists of the set of BG codebook vectors, another set of codebook vectors representing the FG, and mechanisms that update the BG and the FG set of codebook vectors by removing old codevectors and adding new ones in order to obtain an accurate FG segmentation. The model needs to keep information about the current and recent past FG, because static objects in the FG should become BG after a reasonable time.

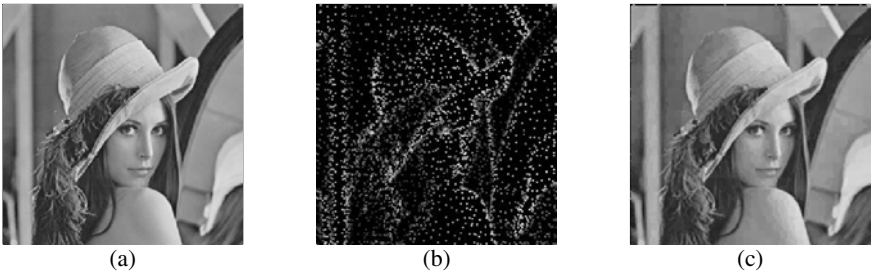
We use two sets of codebook vectors, one for the BG information and the other for the FG. Given an image pixel *x* and the codebooks C_{bg} (background) and C_{fg} (foreground) we define *DualLearn* (Fig. 3), which tells us if and which codebook represents the image pixel, and if none does, it adds the pixel to the FG. With the purpose of managing computational time effectively, we define a seed-growing

[1] In this work the L1 distance was used (for having a faster processing).

strategy for processing a complete image frame based on *DualLearn*. Given a set of image pixels *S*, *DualGrow* (Fig. 4) detects and processes areas where novel image content is found, while skipping areas that remain unchanged. Maintaining an accurate BG set of codebook vectors for an image sequence requires a method for removing codevectors when changes in the BG occur. We introduce a dynamical attribute s_i called score for every codevector c_i. If the codevector represents an image pixel at the codevectors spatial location, the score increases, otherwise it decreases (Figs. 5 and 6). This corresponds to a simple diffusion process using the outcome of the present representation ability as an exogenous input. Codevectors with a score below some threshold are considered obsolete and removed. On the other hand, a maximum score is used to identify static FG.

LearnImage*(I, C)*
foreach $x_j \in I$:
if $!\exists c_i \in C(x_j \in \bar{c}_i)$
$C \leftarrow C \cup \{x_j\}$

Fig. 1. Make *C* to represent an image *I*



(a) (b) (c)

Fig. 2. (a) Picture of Lena. (b) Set of codevectors that represents (a). (c) illustrates the quality of the codevectors by replacing each pixel value by the grey scale value of the codevector that represents (a).

2.3 Skindiff: The Skin Detection Algorithm

Skindiff works in two steps: (i) pixel-wise classification, and (ii) spatial diffusion (see fig. 7). The pixel-wise classification calculates the pixel skinness using a MoG model. The spatial diffusion refines the pixel-wise skin detection result, by taking into consideration neighborhood information for determining smooth skin segments, in which at least one pixel with a large skinness is present (see description in [10]).

One could think that the use of a diffusion process for video skin detection is slow; however this is not the case. Thanks to the use of LUTs (look-up-tables) for implementing the pixel-wise classification and a stack for avoiding a slow recursion in the diffusion process, plus the application of the diffusion algorithm in the codebook vectors (when processing video sequences), a very fast processing is obtained. Depending on its parameterization, the algorithm runs at 15-50 frames per second on a 1GHz Pentium 3 processor.

DualLearn(x, C_{bg}, C_{fg}):

if $\exists c_{i1} \in C_{bg}\,(x \in \bar{c}_{i1})$

return c_{i1}

else if $\exists c_{i2} \in C_{fg}\,(x \in \bar{c}_{i2})$

return c_{i2}

else

$C_{fg} \leftarrow C_{fg} \cup \{x\}$

Fig. 3. Make C_{fg} represent x if C_{bg} does not

DualGrow (I, C_{bg}, C_{fg}):

$S \leftarrow SampleSpace(I)$

while $S \neq \phi$

$x \leftarrow PopItem(S)$

$y \leftarrow DualLearn(x, C_{bg}, C_{fg})$

if $x == y$: // x was added to the foreground

foreach $x_n^s \in ConnectedNeighbors(x^s)$:

$PushItem(S, (x_n^s, I(x_n^s)))$

Fig. 4. Make C_{fg} *and* C_{bg} represent S and connected regions. *ConnectedNeighbors*(x^s) *returns the neighbors of* x^s in the two-dimensional lattice space of the image

UpdateForeground(I, C_{bg}, C_{fg}):

foreach $c_i \in C_{fg}$:

$x \leftarrow (c_i, I(c_i^s))$

if $x \in \bar{c}_i$

$s_i \leftarrow s_i(1-\tau)+\tau$

if $s_i \geq s_{static}$

$C_{bg} \leftarrow C_{bg} \cup \{c_i\}$

$C_{fg} \leftarrow C_{fg} \setminus \{c_i\}$

else

$C_{fg} \leftarrow C_{fg} \setminus \{c_i\}$

Fig. 5. Check for static objects in the foreground

UpdateBackground (I, C_{bg}):

foreach $c_i \in C_{bg}$:

$x \leftarrow (c_i, I(c_i^s))$

if $x \in \bar{c}_i$

$s_i \leftarrow s_i(1-\tau)+\tau$

else

$s_i \leftarrow s_i(1-\tau)-\tau$

if $s_i \leq s_{death}$

$C_{bg} \leftarrow C_{bg} \setminus \{c_i\}$

Fig. 6. Check for removal of background

2.4 Detecting Skin in Videos

All modules involved in the system are integrated in the *RecallImage* algorithm, which is computed at each frame, maintaining the BG/FG models and detecting the skin areas. *RecallImage* has the following processing step (see fig. 8):

A.Initialization: Before starting to process the frames, all sets are initialized. For the first frame, which is considered to be FG, a detailed representation is obtained. The first frame could also be considered as BG, in which case *skindiff* should be run on this set before start calling *RecallImage*, otherwise skin areas appearing on the first frame will be wrongly classified.

B.RecallImage:

1. The frame model is updated: let BG & FG vectors "represent" the image pixels).

2. Prototypes to be used in the diffusion are selected and added to S_{skin}. They

correspond to all prototypes of the FG and prototypes of the BG that were previously classified as skin. It is important to add prototypes of the BG to S_{skin}, otherwise static or slowly moving skin areas would be classified as non-skin and the diffusion would not work, because skin areas are usually homogenous and the diffusion needs "connected" areas to work. In fig. 9(c), FG pixels are sparse, and running the diffusion only on this set would give bad results. However when we consider also the BG codevectors previously classified as skin we obtain good results (see fig. 9(d)).

3. S*kindiff* is applied to the set S_{skin}. C_{skin}, contains the detected skin pixels.

4. Corresponding FG prototypes that were classified as skin are labeled. It is important to notice that a prototype can get the label *Skin* only if it is part of the FG. The label of the BG prototypes is never changed and a BG prototype will have a label *Skin* only if it got it when it was a FG prototype.

5. Static objects in the FG are learnt by the BG model.

6. Too old, non representative, BG vectors are discarded.

```
Skindiff (IM)
  {S_seed, S_min} ← find_seeds(IM)
  foreach ℑ ∈ S_seed
      PushItem(S_skin, ℑ )
      PushItem(Stack, ℑ )
      while(Stack ≠ φ)
          s ← PopItem(Stack)
          foreach x_j^s ∈ ConnectedNeighbors(s^s)
          if x_j ∉ S_skin AND x_j ∈ S_MIN
              if |x_j^r − s^r|^2 < T_tdiff^2
                  PushItem(S_skin, x_j)
                  PushItem(Stack, x_j)
  return S_kin
```

```
find_seeds(IM) {
  foreach x_i ∈ IM
    if g(x_i^r) > T_seed
        PushItem(S_seed, x_i)
    else if g(x_i^r) > T_min
        PushItem(S_min, x_i)
  return {S_seed, S_min}
}
```

IM: Set of prototypes representing the image.
S_{skin}: Final set of skin pixels, output of the algorithm.
S_{min}: Set of pixels that may be skin, S_{seed}: Set of seed pixels.
Stack: Stack data structure.
PopItem(S) removes and returns the first element from the ordered set *S*.
PushItem(S,x) adds *x* to the end of *S*.
ConnectedNeighbors(*s*) returns the neighbors of *s* in the 2D lattice of the image.
g() corresponds to the MoG described on [3] and it is implemented using a LUT [10].

Fig. 7. Skin detection algorithm

3 Results and Analysis

For testing the performance of the proposed algorithm, video sequences captured in our lab or obtained from Internet were used. The selected videos are considered difficult to segment; they have either changing lighting conditions or complex

backgrounds containing surfaces or static and moving objects with skin-like colors. The dataset consist of a total 5882 frames. A ground-truth was generated for about 0.35% of these frames. We are working on generating a larger testing dataset to obtain a better characterization of our system. The more complete dataset and their ground truth will be made available for future studies. We think that the here presented results show the potential of our system. Figure 9 and 10 show results for the segmentation of one frame of the sequence A and one frame of the sequence B. Sequence A has changing lighting conditions and in the sequence B different people enters and leaves a room.

The performance of the system and the effects of the different parameters are analyzed on hand of operation points defined by true positives (TP) and false positives (FP). TP are skin pixels correctly classified as skin and FP are non-skin pixels classified as skin. Figure 11 shows a cloud of operation points for the proposed algorithm for the sequences A and B. We are planning to perform a more exhaustive analysis considering all system parameters, for characterizing their effect on the detection rates, false positive rates, processing time and FG/BG segmentation.

Initialization: $S_{skin} \leftarrow \emptyset,\ C_{skin} \leftarrow \emptyset, C_{bg} \leftarrow \emptyset,\ C_{fg} \leftarrow \emptyset$

$LearnImage(I, C_{fg})$ // or $LearnImage(I, C_{bg})$; $C_{skin} \leftarrow \text{skindiff}(C_{bg})$

RecallImage: // *process image frame* I

1. $DualGrow(I, C_{bg}, C_{fg})$ //The model is updated to represent the frame
2. $S_{skin} \leftarrow \emptyset, C_{skin} \leftarrow \emptyset$ //Vectors to be used in the diffusion are selected:

   $foreach\ c_i \in C_{fg}$: $S_{skin} \leftarrow S_{skin} \cup \{c_i\}$

   $foreach\ c_i \in C_{bg} \mid c_i^{Label} == Skin$: $S_{skin} \leftarrow S_{skin} \cup \{c_i\}$
3. $C_{skin} \leftarrow \text{skindiff}(S_{skin})$
4. $foreach\ a_i \in C_{fg}$:

   $if\ \exists c_i \in C_{skin}(a_i \in \bar{c}_i)$: $a_i^{Label} \leftarrow Skin$
5. $UpdateForeground(I, C_{bg}, C_{fg})$
6. $UpdateBackground(I, C_{bg})$

Fig. 8. Proposed video skin detection algorithm

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

Fig. 9. (a): Frame 271 of the sequence A. (b): Ground-truth of (a). (c): Detected Foreground. (d) Detected skin (white: foreground skin, grey: background skin)

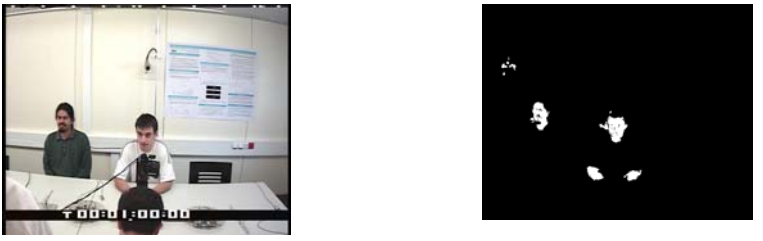


Fig. 10. Left: frame 11100 of sequence B. Right: detected skin in frame 11100

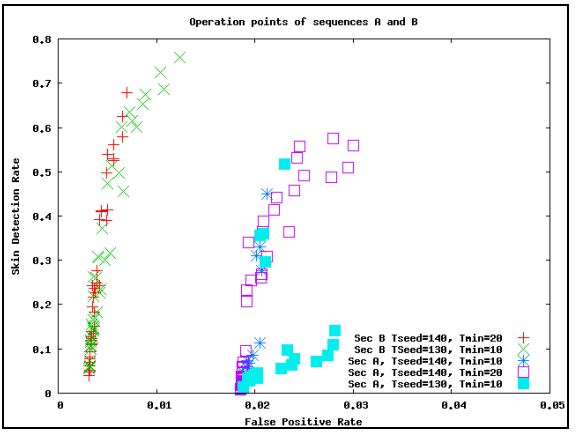


Fig. 11. Operation points of the proposed algorithm for 2 sequences

References

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

[2] B. Jedynak, H. Zheng, and M. Daoudi, "Statistical Models for Skin Detection", IEEE Workshop Statistical Analysis in Computer Vision, together with CVPR 2003.

[3] 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.

[4] 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.

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

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

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

[8] 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.

[9] 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*, Seoul, Korea, 37-42, 2004.

[10] J. Ruiz-del-Solar, J. and R. Verschae, "Robust Skin Detection using Neighborhood Information", *Int. Conf. on Image Processing*, October 24 – 27, Singapore, October 2004.

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