A New Fast Skin Color Detection Technique

Tarek M. Mahmoud

Abstract**—**Skin color can provide a useful and robust cue for human-related image analysis, such as face detection, pornographic image filtering, hand detection and tracking, people retrieval in databases and Internet, etc. The major problem of such kinds of skin color detection algorithms is that it is time consuming and hence cannot be applied to a real time system. To overcome this problem, we introduce a new fast technique for skin detection which can be applied in a real time system. In this technique, instead of testing each image pixel to label it as skin or non-skin (as in classic techniques), we skip a set of pixels. The reason of the skipping process is the high probability that neighbors of the skin color pixels are also skin pixels, especially in adult images and vise versa. The proposed method can rapidly detect skin and non-skin color pixels, which in turn dramatically reduce the CPU time required for the protection process. Since many fast detection techniques are based on image resizing, we apply our proposed pixel skipping technique with image resizing to obtain better results. The performance evaluation of the proposed skipping and hybrid techniques in terms of the measured CPU time is presented. Experimental results demonstrate that the proposed methods achieve better result than the relevant classic method.

Keywords**—**Adult images filtering, image resizing, skin color detection, YcbCr color space.

I. Introduction

SKIN detection is a very popular and useful technique for detecting and tracking human-body parts. It receives much attention mainly because of its wide range of applications such as, face detection and tracking, naked people detection, hand detection and tracking, people retrieval in databases and Internet, etc. The main goal of skin color detection or classification is to build a decision rule that will discriminate between skin and non-skin pixels. Identifying skin colored pixels involves finding the range of values for which most skin pixels would fall in a given color space. In general, a good skin color model must have a high detection rate and a low false positive rate. That is, it must detect most skin pixels while minimizing the amount of non-skin pixels classified as skin. Commonly used skin detection algorithms can detect skin regions accurately. A comprehensive survey on skin detection algorithms can be found in [13], [20].

The major problem of such kinds of algorithms is that it is time consuming and hence cannot be applied to a real time system. Generally, Skin color detection techniques need a long CPU time. Many existing articles deal with speeding up the detection process [8], [17].

Image resizing (resampling) is a standard tool in many image processing applications. It works by uniformly resizing the image to a target size. Recently, there is a growing interest in image retargeting that seeks to change the size of the image while maintaining the important features intact [14], [5]. Changing the size of the image has been extensively studied in the field of texture synthesis, where the goal is to generate a large texture image from a small one. Efros et al. [1] find seams that minimize the error surface defined by two overlapping texture patches. This way, the original small texture image is quilted to form a much larger texture image. This was later extended to handle both image and video texture synthesis by Kwatra et al. [19] that showed how to increase the space and time dimensions of the original texture video. Many fast skin color detection techniques are based on the image resizing [12]. To overcome the time consuming problem in skin color detection, we introduce a new fast technique for skin detection which can be applied in a real time system. In this technique, instead of testing each image pixel to label it as skin or non-skin (as in classic techniques), we skip a predetermined number of pixels. The reason of the skipping process is the high probability that neighbors of the skin color pixels are also skin pixels, especially in adult images and vise versa. The skipping process can be applied with the resizing technique to obtain better results. This hybrid technique takes the advantages of both methods.

The paper presents some experimental results to evaluate the performance of the proposed skipping and hybrid techniques in terms of the measured CPU time.

The remainder of the paper is organized as follows: section 2 describes the YcbCr color space. The image resizing technique is given in section 3. In section 4, the proposed skipping and hybrid techniques are given. Experimental results used to evaluate the performance of these techniques are described in section 5. Finally, our conclusion and future work are presented.

II. The YC_RC_B Color Space

The choice of color space can be considered as the primary step in skin-color classification. The RGB color space is the default color space for most available image formats. Any other color space can be obtained from a linear or non-linear transformation from RGB. The color space transformation is assumed to decrease the overlap between skin and non-skin pixels thereby aiding skin-pixel classification and to provide robust parameters against varying illumination conditions Although there exist many color spaces, we opt for YC_rC_b

Tarek M. Mahmoud is with the Computer Science Dept., Faculty of Science, Minia University, El-Minia, Egypt; e-mail: tarek_2ms@yahoo.com.

International Science Index, Computer and Information Engineering Vol:2, No:7, 2008 waset.org/Publication/14580

color space because its effectiveness in skin detection has been shown previously [15]. YC_rC_b is an encoded nonlinear RGB signal, commonly used by European television studios and for image compression work. Color is represented by luma (which is luminance, computed from nonlinear RGB [4], constructed as a weighted sum of the RGB values, and two color difference values C_r and C_b that are formed by subtracting luma from RGB red and blue components.

$$Y = 0.299\ R + 0.587\ G + 0.114\ B$$

$$C_r = R - Y$$

$$C_b = B - Y$$

While YC_rC_b is device dependent, it is intended for use under strictly defined conditions within closed systems. The Y component describes brightness, the other two values describe a color difference rather than a color, making the color space unintuitive. The transformation simplicity and explicit separation of luminance and chrominance components makes this color space attractive for skin color modeling [2], [3], [6], [11], [16], [18].

YC_rC_b was developed as part of the ITU-R Recommendation B.T. 601 for digital video standards and television transmissions. It is a scaled and offset version of the Y UV color space. In YC_rC_b, the RGB components are separated into luminance (Y), chrominance blue (C_b) and chrominance red (C_r). The Y component has 220 levels ranging from 16 to 235, while the C_r, C_b components have 225 levels ranging from 16 to 240:

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.4810 & 128.5530 & 24.9660 \\ -37.7745 & -74.1592 & 111.9337 \\ 111.9581 & -93.7509 & -18.2072 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where the R, G,B values are scaled to [0,1].

In contrast to RGB, the YCbCr color space is luma-independent, resulting in a better performance. The corresponding skin cluster is given as [9]:

$Y > 80$
$85 < Cb < 135$
$135 < Cr < 180$,
where $Y, Cb, Cr = [0,255]$.

Chai and Ngan [7] have developed an algorithm that exploits the spatial characteristics of human skin color. A skin color map is derived and used on the chrominance components of the input image to detect pixels that appear to be skin. The algorithm then employs a set of regularization processes to reinforce those regions of skin – color pixels that are more likely to belong to the facial regions. Working in the YC_rC_b space Chai and Ngan have found that the range of C_b and C_r most representative for the skin – color reference map were:

$$77 \le C_b \le 127 \quad and \quad 133 \le C_r \le 173$$

III. Image Resizing Technique

Image resizing (or resampling) is one of the most common functions of every raster image processing tool. Graphics Device Interface GDI+ is an improved 2D graphics environment, adding advanced features such as anti-aliased 2D graphics, floating point coordinates, gradient shading, more complex path management, intrinsic support for modern graphics-file formats like JPEG and PNG (which were conspicuously absent in GDI). If you've decided for GDI+ to resize your images, you can choose from variety of filters. Libor Tinka [10] implemented a collection of different resampling filters. Some of them provide even better results than filters in professional imaging applications. He used C#.net to implement the Resampling Service. Some of the used filters in this implementation are shown in Table I.

TABLE I A COLLECTION OF DIFFERENT RESAMPLING FILTERS

Filter	Description
Box	equivalent to Nearest Neighbor on upsampling, averages pixels on downsampling
Triangle	equivalent to Low; the function can be called Tent function for its shape
Hermite	use of the cubic spline from Hermite interpolation
Bell	attempt to compromise between reducing block artifacts and blurring image
CubicBSpline	most blurry filter (cubic Bezier spline) - known samples are just "magnets" for this curve
Lanczos3	windowed Sinc function (sin(x)/x) - promising quality, but ringing artifacts can appear
Mitchell	another compromise, but excellent for upsampling
Cosine	an attemp to replace curve of high order polynomial by cosine function (which is even)
CatmullRom	Catmull-Rom curve, used in first image warping algorithm (did you see Terminator II ?)
Quadratic	performance optimized filter - results are like with B-Splines, but using quadratic function only
Quadratic BSpline	quadratic Bezier spline modification
Cubic Convolution	filter used in one example, its weight distribution enhances image edges
Lanczos8	also Sinc function, but with larger window, this function includes largest neighborhood

The skin color detection with resizing technique can be summarized as follows:

Step 1. Extract the images from web pages.
Step 2. Determine the used filter for resizing.
Step 3. Apply the resizing method with the extracted image, image width/2, and image height/2 as arguments.

Step 4. Apply the skin color detection on the output image after resizing.
Step 5. Record the CPU time of the detection process for both the classic and resized techniques.

IV. The Proposed Skipping Technique

Commonly used skin detection algorithms can extract skin regions from images accurately and reliably, but they often take a long CPU time to finish the detection process. As described in section 3, the resizing algorithm can be considered as fast skin region detector technique. In all methods used in skin color detection, each image pixel is tested and labeled as skin or non-skin pixel. In the proposed method, instead of testing each image pixel, we skip vertically/horizontally a predetermined number of pixels. The reason of the skipping process is the high probability that neighbors of the skin color pixels are also skin pixels, especially in adult images and vise versa. The steps of the proposed method can be described as follows:

Step 1. Extract the images from web pages.
Step 2. Apply the skipping method on the extracted images.
Step 3. Apply the skin color detection on the input image with skipping a predetermined number of pixels.
Step 4. Record the CPU time of the detection process for both the classic and skipping techniques.

We can combine the proposed method with the resizing method to obtain better results. This hybrid technique takes the advantages of both methods. In this technique, we first resize the extracted image, then we apply the skipping technique in the skin color detection step. The hybrid algorithm can be described as follows:

Step 1. Extract the images from web pages.
Step 2. Determine the used filter for resizing.
Step 3. Apply the resizing method with the extracted image, image width/2, and image height/2 as arguments.
Step 4. Apply the skin color detection on the resized image.
Step 5. Apply the skipping method on the resized image.
Step 6. Record the CPU time of the detection process for both the classic and hybrid techniques.

5 Experimental Results

Finally, we present a comparison between the two proposed fast skin recognition techniques (skipping and hybrid) described in section 4 with the classic technique. Each algorithm is implemented using C# programming language. The experiments are done on a 2.66 GHz Pentium IV PC running Microsoft Windows XP operation system. We present tables showing the extracted image number, image width, image height, and CPU time (measured in ms) of each classic method, resizing method, proposed method, and hybrid method. Each technique is applied on a wide variety of images extracted from web pages. To illustrate the accuracy of the proposed skipping technique, we apply it and the hybrid one on variety of extracted images with different pixels skipping rate.

Tables II, III, and IV illustrate the CPU time for all techniques with skipping rates 3, 10, and 15 pixels respectively. The corresponding skin detection with the same skipping rates is displayed in Figures 1, 2, and 3.

Tables II Measured the CPU time in ms for all techniques with skipping rate 3 pixels

Image No.	Width	Height	Classic Method	Proposed Method	Resizing Method	Hybrid Method
1	333	500	14.062	9.547	4.343	2.688
2	500	333	16.141	11.422	4.719	2.984
3	500	375	16.984	11.141	5.187	3.422
4	500	375	30.14	19.891	8.437	6
5	500	351	36.485	34.234	10.812	7.86
6	500	353	31.375	21.094	7.625	5.812
7	500	375	35.797	24.5	9.641	6.578
8	500	375	34.797	22.125	7.813	5.453
9	500	375	27.375	17.281	7.328	4.579
10	500	375	24.688	14	6.156	3.562
11	358	500	26.297	16.125	6.703	4.266
12	500	375	31.281	20.735	7.797	5.656
13	500	375	26.703	16.953	6.922	4.453
14	339	500	25.594	16.844	6.562	4.563
15	357	500	26.657	17.921	7.36	4.75
16	230	340	12.937	8.875	3.281	2.25
17	200	250	7.375	4.718	1.938	1.297
18	200	250	6.297	3.5	1.594	0.922
19	200	250	7.719	4.5	1.859	1.203
20	200	250	6.813	4.062	2.078	1.266
21	200	250	7	4.281	1.844	1.187
22	200	250	7.344	4.671	1.922	1.282
23	200	250	8.562	6.172	2.203	1.594
24	200	250	7.5	5	2.062	1.407

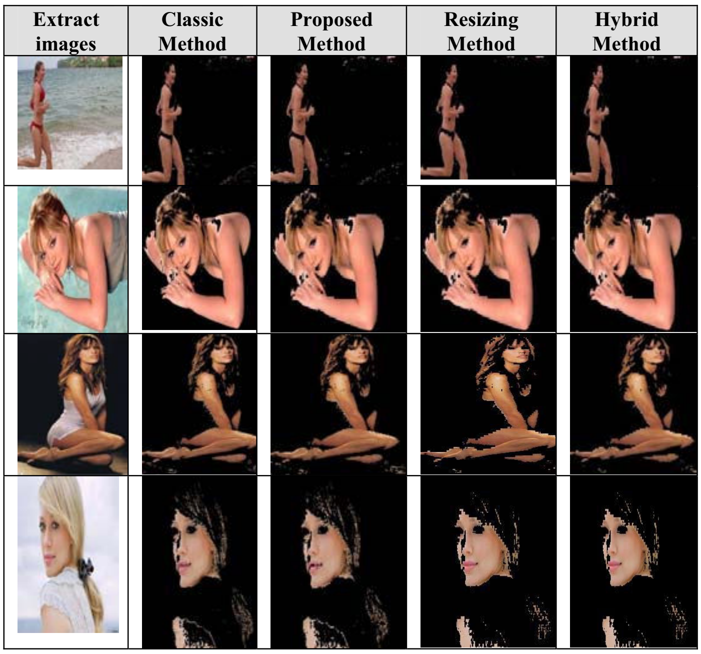

Fig. 1 The corresponding skin detection with skipping rate 3 pixels

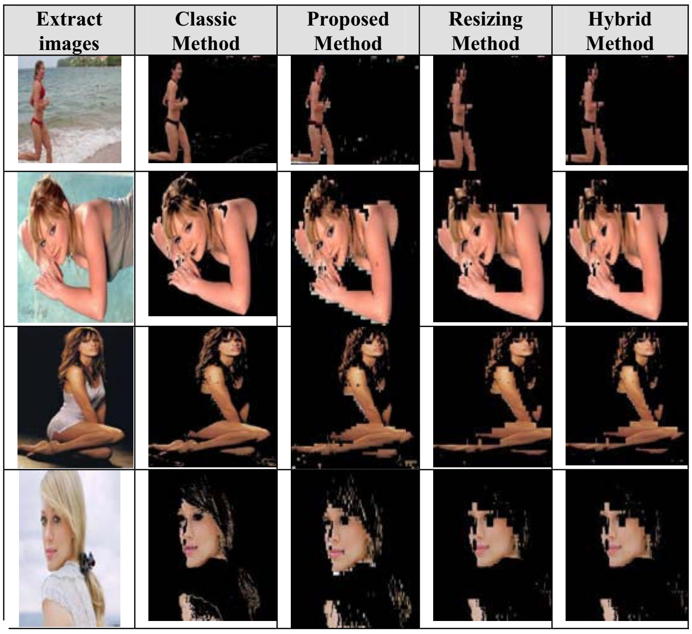

Fig. 2 The corresponding skin detection with skipping rate 10 pixels

TABLES III MEASURED THE CPU TIME IN MS FOR ALL TECHNIQUES WITH SKIPPING RATE 10 PIXELS

Image No.	Width	Height	Classic Method	Proposed Method	Resizing Method	Hybrid Method
1	333	500	14.875	7.61	4.062	2.063
2	500	333	15.312	8.469	4.39	2.329
3	500	375	15.422	7	4.156	1.953
4	500	375	20.437	11.172	5.656	2.985
5	500	351	21.187	11.594	5.75	3.313
6	500	353	24.109	13.312	6.204	3.468
7	500	375	25.39	13.688	6.578	3.641
8	500	375	24.765	12.969	6.5	3.516
9	500	375	22.046	10.422	5.688	2.75
10	500	375	19.656	8.031	5.328	2.016
11	358	500	21.078	10.516	5.968	2.969
12	500	375	28.016	13.812	6.531	3.703
13	500	375	21.985	10.343	6.047	2.813
14	339	500	21.078	10.875	5.391	2.812
15	357	500	22.781	11.25	5.938	3.047
16	230	340	10.485	5.531	3.281	1.61
17	200	250	6.625	3.093	1.86	0.89
18	200	250	5.704	1.968	1.61	0.578
19	200	250	6.922	2.906	1.672	0.844
20	200	250	6.171	2.454	1.531	0.703
21	200	250	6.391	2.609	1.563	0.812
22	200	250	6.5	2.89	1.781	0.844
23	200	250	7.547	4.11	3	1.187
24	200	250	7.36	3.281	1.625	0.984

TABLES IV MEASURED THE CPU TIME IN MS FOR ALL TECHNIQUES WITH SKIPPING RATE 15 PIXELS

Image No.	Width	Height	Classic Method	Proposed Method	Resizing Method	Hybrid Method
1	333	500	16.985	9.453	5.625	2.984
2	500	333	29.422	19.093	10.204	5.031
3	500	375	39.531	15.344	9.735	4.031
4	500	375	47.297	22.937	12.5	7.828
5	500	351	38.968	23.844	11.703	5.781
6	500	353	46	24.25	12	6.141
7	500	375	46.625	22.422	11	6.234
8	500	375	43.515	21	11.328	5.922
9	500	375	38.25	16.438	10.25	4.453
10	500	375	34.234	12.469	8.485	3.25
11	358	500	36.359	15.828	9.25	4.344
12	500	375	43.609	22.172	12.594	7.25
13	500	375	44.157	18.703	12.187	7.907
14	339	500	45.547	18.922	9.484	5
15	357	500	41.422	19.844	10.031	5.437
16	230	340	20.032	11.14	5.578	2.735
17	200	250	13.328	6.172	3.156	1.719
18	200	250	10.828	3.718	3.016	1.125
19	200	250	12.281	5.547	2.782	1.328
20	200	250	12.844	4.125	2.906	1.125
21	200	250	13.015	5.328	3.829	1.171
22	200	250	13.407	4.812	2.859	1.282
23	200	250	15.625	7.704	3.593	1.891
24	200	250	11.422	4.641	2.89	1.375
Total			**714.703**	**335.906**	**186.985**	**95.344**

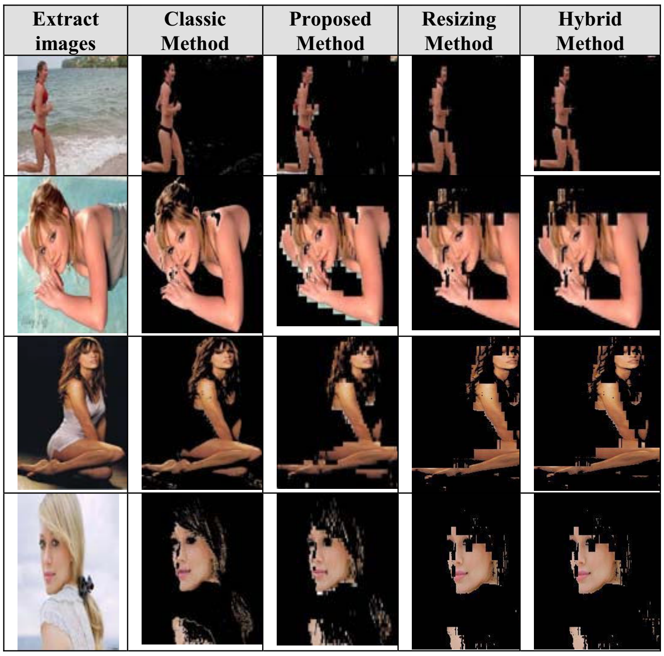

Fig. 3 The corresponding skin detection with skipping rate 15 pixels

As can be seen in Tables II, III, IV and Figures 1, 2, and 3:

1- Using the skipping and resizing techniques speed up the required CPU time to extract skin color regions compared with the classic method.

2- Increasing the skipping rate reduces the CPU time, but the extracted skin color regions are inaccurate. Therefore, we have to select the pixels skipping rate such that it does not affect the accuracy of the skin color detection.

3- The average saving CPU time of the proposed skipping technique compared with the classic technique is approximately 53.00%.
4- The average saving CPU time of the resizing technique compared with the classic technique is approximately 73.84%.
5- The average saving CPU time of the proposed hybrid technique compared with the classic technique is approximately 86.65%.
6- The proposed hybrid technique has the best performance compared with all techniques.

Accordingly, the proposed hybrid technique can be used in any filtering system based on skin color detection to prevent adult images from displaying.

V. CONCLUSION

This paper suggests two new techniques that can be used to greatly improve the CPU time of skin color detection algorithms. The first technique uses an image pixel skipping process instead of testing each pixel to label it as skin or non-skin. The second technique combines image resizing with the skipping techniques. The performance evaluation of the proposed skipping and hybrid techniques in terms of the measured CPU time is presented. Experimental results demonstrate that the proposed methods achieve better result than the relevant classic method. According to the experimental results, the two proposed techniques are suitable for the extracted adult images from web pages. So, the proposed hybrid technique can be used in any filtering system based on skin color detection to prevent adult images from displaying. In the future work, the determination of skipping rate will be studied to obtain an accurate skin color detection and fast CPU time.

REFERENCES

[1] A., Efros, and W. Freeman, “Image quilting for texture synthesis and transfer”. In SIGGRAPH’01, 341–346, 2001.

[2] B.D., Zarit, B.J., Super, and F.K.H. Quek, "Comparison of five color models in skin pixel classification". In Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pages 58-63, Corfu, Greece, Sept. 1999.

[3] B.Menser, and M. Wien, "Segmentation and tracking of facial regions in color image sequences". In SPIE Visual Communications and Image Processing, Perth, Australia, 2000.

[4] C. A. Poynton,. "Frequently asked questions about colour". Available at ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz. 1995.

[5] C., Wanga, P., Xueb, and W., Lin, “Layered image resizing in compression domain”, Signal Processing: Image Communication 23 (2008) 58–69. Available at www.sciencedirect.com

[6] D. Chai, and A., Bouzerdoum, “A Bayesian Approach to Skin Color Classification in YCbCr Color Space”. In Proc. Of IEEE Region Ten Conference, vol. 2, 421- 4124, 1999.

[7] D. Chai, and K.N. Ngan, "Face segmentation using skin-color map in videophone applications". IEEE Trans. on Circuits and Systems for Video Technology, 9(4): 551-564, June 1999.

[8] F. Tomaz, T. Candeias, and H. Shahbazkia, "Fast and Accurate Skin Segmentation in Color images". CRV04, 2004.

[9] G. Kukharev, A. Novosielski, “Visitor identification elaborating real time face recognition system”, In Proc. 12th Winter School on Computer Graphics (WSCG), Plzen, Czech Republic, pp. 157-164, Feb. 2004.

[10] “Image Resizing - outperform GDI+”, URL address: http://www.codeproject.com/KB/GDI-plus/imgresizoutperfgdiplus.aspx

[11] J. Ahlberg,. "A system for face localization and facial feature extraction". Tech. Rep. LiTH-ISY-R-2172, Linkoping University, 1999.

[12] J. Mukherjee, and S.K. Mitra, “Image resizing in the compressed domain using subband DCT”, IEEE Trans. Circuits Syst. Video Technol. 12 (July 2002) 620–627.

[13] P. Kakumanu, S. Makrogiannis, N. Bourbakis, "A Survey of Skin-Color Modeling and Detection Methods", Pattern Recognition 40, pp 1106-1122, 2007. available at www.sciencedirect.com.

[14] R. Dugad, and N. Ahuja, “A fast scheme for image size change in the compressed domain”, IEEE Trans. Circuits Syst. Video Technol. 11 (April 2001) 461–474.

[15] R., Gonzales, and E., Woods, “Digital Image Processing,” Prentice Hall, Inc, New Jersey, 2002.

[16] R.L., Hsu, M., Abdel-Mottaleb, and A.K. Jain, "Face detection in color images". IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(5):696-706, May 2002.

[17] R. Seguier, “A very fast adaptive face detection system”, in Proc. IASTED Conf. on Visualization, Imaging, and Image Processing (VIIP), Marbella, Spain, Sep. 2004.

[18] S. L., Phung, A., Bouzerdoum, and D. Chai, "A novel skin color model in YCbCr color space and its application to human face detection". In IEEE International Conference on Image Processing (ICIP’2002), vol. 1, 289–292, 2002.

[19] V., Kwatra, A., Schodl, I., Essa, G., Turk, and A., Bobick, “Graphcut textures: Image and video synthesis using graph cuts”. In SIGGRAPH’03, 277–286, 2003.

[20] V., Vezhnevets, V., Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques". In GraphiCon, Moscow, Russia, Sept. 2003.