Robust color segmentation algorithms in illumination variation conditions

Jinhui Lan (蓝金辉)* and Kai Shen (申 凯)

Department of Measurement and Control Technologies, School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China

**E-mail: jh.lan@263.net*

Received June 15, 2009

Changing illumination condition can change the result of image segmentation algorithm and reduce the intelligent recognition rate. A novel color image segmentation method robust to illumination variations is presented. The method is applied to the skin segmentation. Based on the hue preserving algorithm, the method reduces the dimensionality of the red-green-blue (RGB) space to one dimension, while keeping the hue of every pixel unchanging before and after space transformation. In the new color space, the skin color model is established using Gaussian model. Experimental results show that the method is robust to illumination variations, and has low computational complexity.

OCIS codes: 100.2000, 100.3008.

doi: 10.3788/COL20100803.0286.

Color segmentation is an essential, critical, and preliminary process in a lot of vision-based tasks such as visual tracking, human-computer interaction (HCI), human-robot interaction (HRI), vision-based robotics, visual surveillance, and so forth, because color is an effective and robust visual cue for characterizing an object from the others. Recently, there has been a growing interest in the skin segmentation, which aims at detecting human skin regions in an image.

However, changing illumination condition can change the characteristics of a color, and limit the applications of the color segmentation[1]. Therefore, a lot of researches have been carried out for invariant detection of a color under illumination-variation conditions. Static skin model based methods were adopted[2−5], which firstly selected an appropriate color space. The color space transformation has been assumed to increase separability between skin and non-skin classes to increase similarity among different skin tones. Then a model of skin color in the selected color space is established to decide whether a pixel belongs to a skin region. The color space mainly includes normalized red-green (RG)[2], hue-saturation-value (HSV) (hue-saturation-illumination (HSI))[5], YCrCb[6], YIQ[7], YES[8], etc. The skin color model includes single Gaussian model[2], Gaussian mixture models[3], Gaussian mixture models considering intensity information[5], etc. Cho *et al.* and Soriano *et al.* used dynamic learning based methods for color segmentation[9,10]. These methods dynamically allocate a color through various illumination conditions while those models are updated for every image sequence. Adaptive skin color filter and 'skin locus' adaptive skin color modeling are two typical methods of dynamic learning based methods.

In this letter, a novel color image segmentation method which is robust to illumination variations is proposed. Since RGB space is the most common space to represent color images, and HSI space can separate the chrominance and luminance in a color image to a certain extent, we consider both the characteristics of RGB space and HSI space, and reduce the dimensionality of the RGB space to one dimension while preserving the hue of every pixel in the image before and after space transformation constant by hue-preserving algorithm. Therefore, 1D threshold can be used to subdivide the color image. Experimental result verifies that the proposed method is not only robust to the changing illumination condition, but also more efficient in computation efforts.

Hue preservation is necessary when converting colors from one color space to another. Distortion may occur if hue is not preserved. The hues of a pixel in the scene before and after the transformation should be the same for a hue preserving transformation. First of all, a general hue-preserving method between RGB space and HSI space is introduced for color image segmentation.

The conversion equations from the RGB space to HSI space are noted as

$$\begin{cases} H = \arctan\left(\dfrac{\sqrt{3}\,(G-B)}{(R-G)+(R-B)}\right) \\ S = 1 - \dfrac{3\,[\min(R,G,B)]}{(R+G+B)} \\ I = \dfrac{1}{3}(R+G+B) \end{cases}. \quad (1)$$

In general, color images are stored and viewed using RGB space. To process an image for segmentation in HSI space, the image needs to be transformed. As can be seen from Eq. (1), this transformation is computationally costly[11], and the inverse coordinate transformation has to be implemented for displaying the images.

Two hue-preserving operations, scaling and shifting, were introduced in Ref. [12] for luminance and saturation processing. Using these two operations, hue-preserving method for color image segmentation is developed.

We denote the grey values for R, G, and B components of an image pixel I by a vector $\mathbf{x}$, where $\mathbf{x} = (x_1, x_2, x_3)$, x_1, x_2, and x_3 correspond to the red, green, and blue pixel values, respectively. That is, $0 \le x_k \le 255$, $k = 1$, 2, and 3.

Scaling the vector $\mathbf{x}$ to $\mathbf{x}'$ by a factor $\alpha > 0$ is defined as

$$\mathbf{x}' = (x_1 \cdot \alpha, x_2 \cdot \alpha, x_3 \cdot \alpha). \tag{2}$$

Shifting the vector $\mathbf{x}$ to $\mathbf{x}'$ by a factor β is defined as

$$\mathbf{x}' = (x_1 + \beta, x_2 + \beta, x_3 + \beta). \tag{3}$$

A transformation which is a combination of scaling and shifting can be written as

$$\mathbf{x}' = (\alpha x_1 + \beta, \alpha x_2 + \beta, \alpha x_3 + \beta), \tag{4}$$

where x'_k is linear in x_k, $k = 1,\ 2,\ 3$, α and β are not dependent upon $\mathbf{x}$.

It can be seen from Eq. (1) that the transformation, as given in Eq. (4), is hue preserving. Hue preserving means that after transformation the hue of $\mathbf{x}'$ is the same as that of $\mathbf{x}$.

we can define a general transformation as

$$x'_k = \alpha(\mathbf{x}) x_k + \beta(\mathbf{x}), k = 1, 2, 3, \tag{5}$$

where α and β are functions of $\mathbf{x}$. The color vector $\mathbf{x}'$ as defined in Eq. (5) possesses the same hue as the color vector $\mathbf{x}$ for any two functions α and β. Next, we select an appropriate color space for robust color segmentation. Then, using hue-preserving method, we study the two functions α and β involved in Eq. (5) for obtaining the mathematical expression of color space transformation from RGB space to the space selected by us.

Generally, the hue and saturation in HSI color space are not affected by brightness. However, experiments verified the fact that color distribution of a single-colored object was not invariant with respect to brightness variations even in the hue-saturation (HS) plane[3]. Additionally, the R, G, and B components of a pixel are closely related in the RGB space. It is difficult to do the segmentation in RGB space which is robust to brightness variations. The connecting line which connects the three corners of the RGB cube is chosen as the color segmentation space. The coordinates of the corners are $R(255, 0, 0)$, $G(0, 255, 0)$, $B(0, 0, 255)$. Firstly, because of the inter-restricted relationship among the R, G, and B components, this connecting line can be regarded as a one-dimensional (1D) space so that the threshold can be set conveniently. Secondly, it is the edge of a HS plane in the HSI space and includes all kinds of hues, the saturation of which is 1. So every kind of hue which we care about can be extracted in this space. Finally, all the intensities of the colors included in it are 255/3. Consequently, this area is robust to brightness variations.

Based on the hue-preserving method, the color vector which has the same hue in the RGB space can be mapped to a point on the connecting line fast. The mapping steps are as follows: 1) Adjusting the intensity of every pixel to k. According to Eq. (2), while R, G, and B change at the same ratio, the intensity of a pixel changes, but the hue and saturation keep constant. Let vector $\mathbf{y}=\left(\hat{R}, \hat{G}, \hat{B}\right)$ denote the color of the pixel after being adjusted:

$$\mathbf{y} = \frac{k}{p}\mathbf{x}, \tag{6}$$

where k is the intensity value of the pixel which we want to adjust to, $k = \frac{255}{3}$; p is the original intensity of a pixel, $p = (R + G + B)/3$; $\mathbf{x} = (R,\ G,\ B)$ denotes the original pixel's color. 2) Adjusting the saturation (S) of every pixel to 1. According to Eq. (3), adding a constant to R, G, and B at the same time, the intensity and saturation of a pixel change, but the hue keeps constant. Therefore,

$$S = 1 - \frac{3\left[\min\left(\hat{R}, \hat{G}, \hat{B}\right) + t\right]}{\left(\hat{R} + \hat{G} + \hat{B}\right) + 3t} = 1. \tag{7}$$

Adjusts the saturation of a color to 1, where min $\left(\hat{R}, \hat{G}, \hat{B}\right)$ is the minimum value among $\hat{R}$, $\hat{G}$, and $\hat{B}$. The unknown t added to the R, G, and B components simultaneously is calculated. From Eq. (7), we can obtain

$$t = -\min\left(\hat{R}, \hat{G}, \hat{B}\right). \tag{8}$$

Adding the constant t which is obtained in Eq. (7) to the R, G, and B components, we get a new color whose saturation is 1 as

$$\mathbf{m} = \mathbf{y} + t. \tag{9}$$

3) Adjusting the intensity of every pixel to k again, while keeping hue and saturation unchanging. Let vector $\mathbf{z} = \left(\check{R}, \check{G}, \check{B}\right)$ denote the new color of the pixel after change:

$$\mathbf{z} = \mathbf{m} \times \frac{k}{k + t}. \tag{10}$$

With these steps, all the colors included in the RGB space can be mapped to the curve discussed above while keeping the hue of the color constant. To sum up, the dimensionality reduction process is shown in Fig. 1.

Finally, the process of color space conversion can be summarized in the form of Eq. (5) as

$$x'_k = \alpha(\mathbf{x}) x_k + \beta(\mathbf{x}), k = 1, 2, 3, \tag{11}$$

$$\alpha(\mathbf{x}) = \frac{k^2}{[k - \min(x_k)] \cdot p}, \tag{12}$$

$$\beta(\mathbf{x}) = \frac{k^2 \cdot \min(x_k)}{[k - \min(x_k)] \cdot p}, \tag{13}$$

where $\mathbf{x} = (x_1, x_2, x_3)$ is the color of the pixel before transformation; $\mathbf{x}' = (x'_1, x'_2, x'_3)$ is the color of the pixel after transformation; $k = \frac{255}{3}$; p is the original intensity of a pixel, $p = (R + G + B)/3$.

The curve mentioned above has some geometric features in the RGB space: one of the components of the vector included in the curve must be 0, and the sum of another two components is 255. Although a three-dimensional (3D) threshold is still needed to be set, once the value of one component is decided, another two are decided simultaneously. Consequently, in fact, what needs to calculate is a 1D threshold. After the color space transformation steps, skin color pixels are extracted from

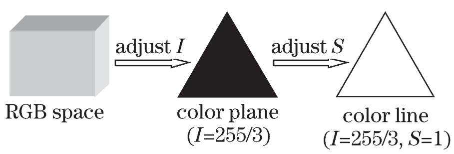

Fig. 1. Dimensionality reduction process.

a lot of processed images, and the skin color model is made in the color line obtained above. The color component chosen as the segmentation parameter can be assumed to form Gaussian distribution because of using a lot of pixels to make the proposed color model. We call x the random variables of the chosen component (e.g.,R). μ_x is the mean value of x. σ_x is the standard deviation of x. The probability density function (PDF) of x has the form

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{14}$$

According to the properties of the Gaussian distribution, the appropriate range of x for image segmentation is from $\mu - 2\sigma$ to $\mu + 2\sigma$.

As discussed above, we make a skin color model in the RGB space using skin color pixels extracted from a lot of processed images. The sampling process of selecting skin pixels is systematic. Firstly, we acquire an image as turning off a lamp to naturally make various illumination conditions in laboratory where several lamps are turned on. Secondly, a total amount of 1000 pixels is manually collected from 20 images with 512 × 768 pixels, acquired under various kinds of illumination conditions. Thirdly, pixels are acquired from several regions of human skin, including forehead, cheeks, nose, neck, and so forth. Finally, skin pixels containing strong highlights and too dark shadows are discarded. The skin color model in the RGB space made by sampling pixels is presented in Fig. 2.

As illustrated in Fig. 2, the curve $R + G = 255$ represents the skin model. The range of R is from 140 to 255, and the range of G is from 0 to 115. The R component is chosen as the parameter to patition images. Calculating μ_R and σ_R, the appropriate range of the 1D threshold is from 150 to 255.

To verify that our method is robust to the brightness variations, we used the experimental picture in Ref. [5]. We conducted experiments for skin color segmentation algorithms respectively belonging to each category mentioned before (the static skin model based method and the dynamic learning based method), and compared the experimental results with those of the proposed method. The two categories include the B-spline curve method[5] and an adaptive classifier[9]. The B-spline curve method has the best performance in the field of skin segmentation until now.

We apply the method proposed to the test images. Figure 3 is the skin image under different illumination conditions. Figures 4(a)–(f) are the segmentation results of the corresponding part of Fig. 3. In Table 1, TPx, FPx, TPo, and FPo mean the number of true (skin) pixels, the number of false (non-skin) pixels, the number of true positive pixels, and the number of false positive pixels, respectively. The evaluation index (EI) for color segmentation expressed is computed by

$$\mathrm{EI} = \omega_1 \times \mathrm{TPR} + \omega_2 \times (1 - \mathrm{FPR}), \tag{15}$$

where $\omega_1 = \omega_2 = 0.5$, EI represents a ratio at which pixels are correctly classified as skin or non-skin, TPR represents the ratio of TPo to TPx, and FPR represents the ratio of FPo to FPx. In Table 1, EIs are computed for the resultant images.

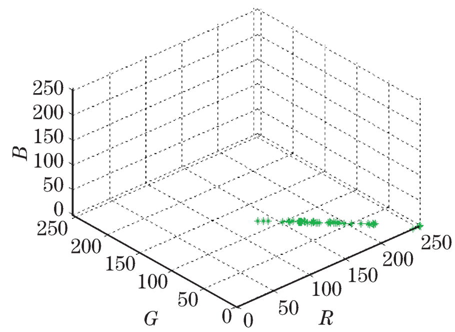

Fig. 2. Skin color model in the RGB space.

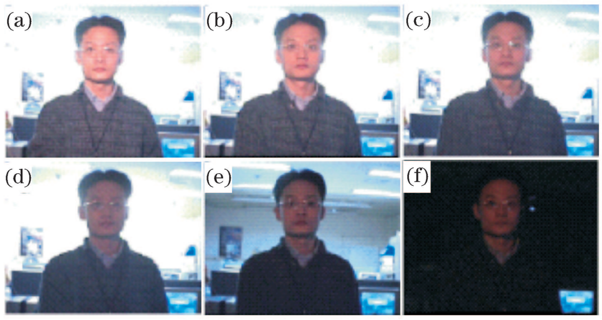

Fig. 3. Skin images under different illumination conditions (test images are reproduced by permission of Professor Kim).

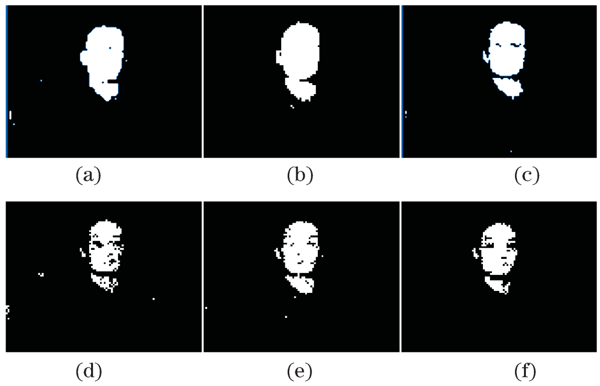

Fig. 4. Segmentation results of Fig. 3.

Table 1. EIs for the Images in Fig. 4 ($\omega_1 = \omega_2 = 0.5$, TPx = 815, FPx = 8257)

	TPo	TPR	FPo	FPR	EI
a	721	0.8847	94	0.01139	0.9367
b	751	0.9215	113	0.01369	0.9539
c	672	0.8245	174	0.02107	0.9017
d	576	0.7067	35	0.00424	0.8512
e	678	0.8319	40	0.00484	0.9976
f	815	0.1000	257	0.03113	0.9844
Mean					0.9376
Variation					0.1619

The Segmentation result of our method shown in Fig. 4 is compared with those of the B-spline curve method and the adaptive classifier method in Fig. 5. As illustrated in Fig. 5, the segmentation result of our method is close to that of B-spline curve method. When lacking of

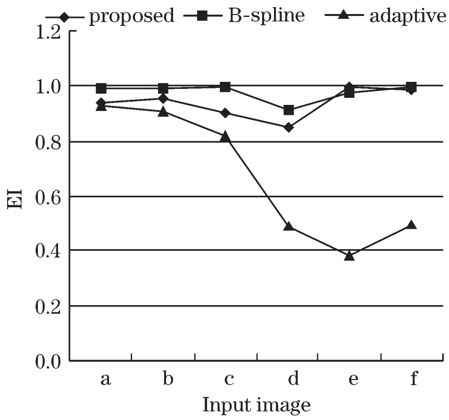

Fig. 5. Comparison of the three different segmentation methods.

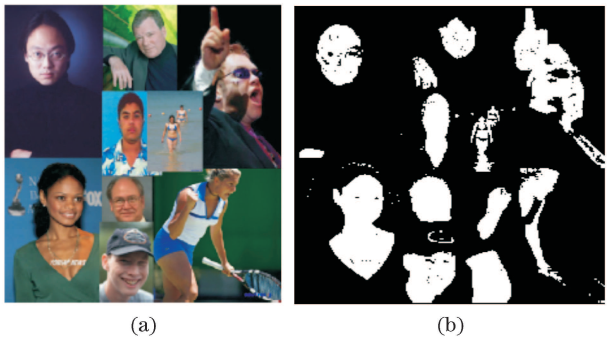

Fig. 6. Experimental results of skin segmentation for the images collected from Internet.

brightness, the result of our method is obviously better than the adaptive classifier method. The mean value of the curve of our method is close to 1, and the variance is low. Consequently, our method is robust to illumination variations.

Experiments for skin color segmentation were conducted using images collected at random from Internet. The experimental results of skin segmentation for these images are shown in Fig. 6. Figure 6(a) is input images while Fig. 6(b) shows resultant images. From Fig. 6, it is also verified that the proposed algorithm is successfully operated for images collected at random from Internet.

The amount of computation of the B-spline curve method and our method were estimated using floating-point multiplication. We subdivided an image with 576×720 pixels, only considering the online computation cost. It takes B-spline curve method $576 \times 720 \times 2$ floating-point operations to do the transformation from RGB space to HSI space, and takes $576 \times 720 \times 10$ operations to decide whether a pixel is in the elliptic region of the skin model[5]. To sum up, it takes $576 \times 720 \times 12$ floating-point operations. By contrast, according to Eqs. (9)–(11), the method proposed by us only takes $576 \times 720 \times 6$ operations.

The processing time of our method and B-spline curve method was evaluated on the digital signal processor (DSP) platform. The core of the platform is a 600-MHz digital media processor (TMS320DM643, Texas Instruments, USA). The size of every frame is 576×720 pixels.

Table 2. Average Processing Time per Frame of the Methods

Segmentation Methods	Average Processing Time per Frame (ms)
Hue-Preserving Method	13
B-Spline Curve Method	42

The optimization option was used to improve the efficiency of the code, which was selected as –o3, –gp. As illustrated in Table 2, the average processing time is about 13 ms per frame by our method; by contrast, the B-spline curve approach takes 42 ms per frame averagely.

Moreover, B-spline curve method needs to do a lot of offline work to build the skin model in every gray scale. When changing the object of segmentation, the workload is huge and complex. The method we proposed only needs to do the transformation in Eqs. (11)–(13) on images of the database, and calculates the mean value and the variation. Therefore, in offline calculation, the algorithm proposed by us greatly reduces the workload; in online computing, this method can reduce the calculation by half.

In conclusion, a color segmentation method which is robust to brightness variations based on the RGB space dimensionality reduction is proposed. Experimental results show that the proposed method can effectively overcome the change of lightness. In terms of skin detection, this method not only achieves the similar result as the B-spline curve method, but also greatly reduces the workload of offline and online computation amount. In the future, this method can be used to extract other objects in illumination changing conditions.

References

1. X. Sun, F. Su, and A. Cai, Acta Opt. Sin. (in Chinese) **11,** 2083 (2008).
2. J. Yang, W. Lu, and A. Waibel, *Skin-Color Modeling and Adaptation* (Springer, Berlin, 1997).
3. S. J. McKenna, S. Gong, and Y. Raja, Pattern Recogn. **31,** 1883 (1998).
4. K. Sobottka and I. Pitas, in *Proceedings of International Conference on Pattern Recognition* **3,** 421 (1996).
5. C. Kim, B.-J. You, M.-H. Jeong, and H. Kim, Pattern Recogn. **41,** 22 (2008).
6. D. Chai and K. N. Ngan, in *Proceedings of Third International Conference on Automatic Face and Gesture Recognition* 124 (1998).
7. Y. Dai and Y. Nakano, Pattern Recogn. **29,** 1007 (1996).
8. E. Saber and A. M. Tekalp, Pattern Recogn. Lett. **8,** 669 (1998).
9. K.-M. Cho, J.-H. Jang, and K.-S. Hong, Pattern Recogn. **34,** 1067 (2001).
10. M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, Pattern Recogn. **36,** 681 (2003).
11. S. K. Naik and C. A. Murthy, IEEE Trans. Image Process. **9,** 505 (2000).
12. C. C. Yang and J. J. Rodriguez, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics* **1,** 667 (1995).