ICICS-PCM 2003
15-18 December 2003
Singapore

A Background Subtraction for a Vision-based User Interface*

Dongpyo Hong and Woontack Woo

KJIST U-VR Lab.
{dhong, wwoo}@kjist.ac.kr

Abstract

In this paper, we propose a robust and efficient background subtraction algorithm for a vision-based user interface. We separate a user of interest as precisely as possible from the acquired images in order to convey the user's intension properly into the system through the vision-based user interface. Although the background subtraction techniques have been adopted in many vision-based interfaces to extract or track moving objects of interest in the images, they still suffer from the changes of lighting, such as shadows and highlighting. The proposed method removes effectively such interferences of lighting changes by exploiting pixel-wise statistical characteristics and threshold values in the well-known two color spaces (RGB and normalized RGB). According to experimental results, the proposed algorithm can be applied to various applications requiring real-time segmentation from the image sequences on the fly.

Keywords: Background Subtraction, Threshold Selection, Color Space.

1. Introduction

In the last few decades, there have been many studies which substitute the conventional user interfaces with new types of interfaces like vision, gesture, voice, and other sensors. Especially, the advantages of a vision-based interface over other sensors are to calibrate easily, interact with the systems naturally and remove the cumbersome devices from users. In general, the vision-based user interfaces are divided into two types [1]. One is the contact vision-based interface which generally uses markers worn by the user. The other is the non-contact vision based interface which generally uses the background subtraction techniques. The contact vision-based interface is able to extract information of interest by simply tracking the markers. However, it has some drawbacks. For example, the contact-based interface requires the user to wear markers. As a result, when markers are occluded or when multiple markers are used, it is hard and error prone to track them [2]. The non-contact vision-based interfaces overcome the limitations of the contact vision-based interfaces. Note that the previous background subtraction techniques which is used in non-contact vision-based user interfaces have exploited the statistical and/or probabilistic color differences between a current image and the reference image, which is trained during a certain period of time or the number of frames [3][4][5]; the speed differences of moving objects [6]; the characteristics of stereo images [7]; and hybrid [8]. The results of these background subtraction techniques are robust and effective enough to apply to the vision-based user interfaces. However, there are some restrictions and complexities to be resolved. For instance, most of them introduce their own complex color models or one representative threshold value [3][4][5]. And motionless users and texture similarities of interests are also difficulties to be used in the vision-based user interface [6][7].

In this paper, we propose a robust and pragmatic background subtraction technique for the vision-based user interface. The proposed method exploits the well-known RGB color space and normalized RGB color space instead of introducing complex color models. Meanwhile, we use each pixel's statistical characteristics in the both color spaces. We assume that each color channel has different characteristics and the characteristics of pixel-wise values over time follow Gaussian distribution [5]. By using these properties, we are able to determine pixel-wise threshold values semi-automatically as a function of mean and standard deviations of the pixels during the background training period. The proposed method reduces the complexities and restrictions of the previous studies. It is a simpler and more pragmatic background subtraction technique for the real-time vision-based user interface.

This paper is organized as follows. In section 2, we explain the detailed algorithm and implementation of the proposed method. In section 3, we show the experimental results and discuss the conclusion in section 4.

2. Background Subtraction Algorithm

The general background subtraction technique is to subtract a current image from the reference image. Although various cues (color, motion, block, etc.) are utilized in many studies, the proposed method exploits the characteristics of the pixel's color values in the well-known two color spaces (RGB and normalized RGB). It is needed to determine the optimal threshold values in the background subtraction techniques. In this section, we explain the properties of each color space and how to determine the pixel-wise optimal threshold values. In

* This work is sponsored by KJIST and ICU Digital Media Lab.

addition, we show how to use the determined threshold values in the proposed algorithm.

2.1 Characteristics of Color Space

The human vision system recognizes the color of objects based on chromaticity and luminance. Inspired by this, we utilize the two well-known color spaces. In the RGB color space, each pixel has both chromaticity and luminance elements. That is, in this color space, two colors are different if either chromaticity or luminance is different. Therefore, when the background subtraction is done in RGB color space, shadow, shade and highlighting are recognized as the user even though they are only different in luminance but almost same in chromaticity. It is difficult to remove these lighting effects from the user only using RGB color space. This issue makes some previous works introduce their own color models which easily exploit chromaticity and luminance [3][4][5]. The separated representation of the chromaticity and luminance in one color model is able to determines each pixel as precisely as possible. However, it requires much complex and expensive computation. In the normalized RGB color space, each pixel has only a chromaticity element. In this color space, we can remove the lighting interferences because they have only luminance differences from the background scene. We exploit the characteristics of the well-known two color spaces instead of introducing a new color model. The proposed method distinguishes the user without cast shadows from the background scene.

2.2 Background Modeling

In the proposed method, we train the background images in RGB and normalized RGB color space, respectively. Then, we can evaluate the mean and standard deviation at pixel i 's (R,G,B) color channels in the reference image during the background training. Each pixel of the reference image is modeled as follows.

$$< \mu_i(R,G,B), \sigma_i(R,G,B), \mu_i(r,g,b), \sigma_i(r,g,b) > \qquad (1)$$

where $\mu_i(R,G,B)$ and $\sigma_i(R,G,B)$ is the vector of the mean and standard deviation of pixel i 's color channels in RGB color space. $\mu_i(r,g,b)$ and $\sigma_i(r,g,b)$ is the vector of the mean and standard deviation of pixel i 's color channels in the normalized RGB color space.

The following equations are show how to compute the vector of the mean and standard deviation at pixel i in RGB and normalized RGB color space.

$$\mu_i(R,G,B) = \frac{1}{N}\sum_{i=0}^{N-1} I_i(R,G,B) \qquad (2)$$

$$\mu_i(r,g,b) = \frac{1}{N}\sum_{i=0}^{N-1} I_i(r,g,b) \qquad (3)$$

$$\sigma_i(R,G,B) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(I_i(R,G,B) - \mu_i(R,G,B))^2} \qquad (4)$$

$$\sigma_i(r,g,b) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(I_i(r,g,b) - \mu_i(r,g,b))^2} \qquad (5)$$

where each $I_i(R,G,B)$ and $I_i(r,g,b)$ represents the vector of pixel i 's color channels in RGB and normalize RGB color space. N is the number of trained images.

2.3 Threshold Selection and Subtraction

When we observe the variations of pixels in the image of static background scene, they are easily modeled as a Gaussian distribution. From this observation, the threshold value of pixel i is mapped by function of standard deviation of pixel i.

$$Th_i(R,G,B) = \alpha \cdot \sigma_i(R,G,B) \qquad (6)$$

$$Th_i(r,g,b) = \beta \cdot \sigma_i(r,g,b) \qquad (7)$$

where $Th_i(R,G,B)$ and $Th_i(r,g,b)$ is threshold value of pixel i in RGB and normalized RGB color space, respectively. α and β is the determinant constant which determines the confidence interval. For example, if $\alpha = \beta = 2$, it has about 95% confidence interval. This determinant constant α and β determine the threshold ranges. Then we simply achieve threshold value at pixel i using standard deviation σ_i by choosing the determinant constants α and β.

Although most of the background subtraction techniques addressed how to determine the threshold values, there are few methods which show the usages of the determined threshold values in the subtraction operations. In the proposed method, we show how to effectively use the determined threshold values to subtract the user from background scene. Equation (8) and (9) is the determinant function which compares the color channels' differences of pixel i and the determined threshold values in RGB and normalized RGB color space, respectively.

$$F_i = u(D_i(R) - Th_i(R)) + u(D_i(G) - Th_i(G)) + u(D_i(B) - Th_i(B)) \qquad (8)$$

$$f_i = u(D_i(r) - Th_i(r)) + u(D_i(g) - Th_i(g)) + u(D_i(b) - Th_i(b)) \qquad (9)$$

$$D_i(x) = I_i(x) - \mu_i(x) \qquad (10)$$

where F_i ($0 \le F_i \le 3$) and f_i ($0 \le f_i \le 3$) are the determinant functions which characterize pixel i in each color space. Here u is an unit step function and it has either 0 or 1. $D_i(x)$ is the vector difference between current image and reference image at pixel i in RGB color space and normalized RGB color space. Thus, if $D_i(x) > Th_i(x)$, then it is 1. Otherwise, it is 0.

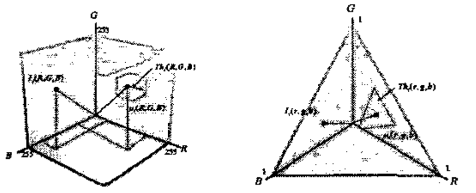
Using equation (8) and (9), we can determine pixel i as follows.

$$I_i = \begin{cases} B: & 0 \le F_i < c_1 \\ H^s: & F_i \ge c_1 \\ B^s: & 0 \le f_i < c_2 \\ H: & f_i \ge c_2 \end{cases} \tag{11}$$

where B is the background image and B^s is the background image with cast shadows. H^s is the segmented user image with shadows and H is the segmented user image without shadows. In RGB and normalized RGB color space, its range is $0 \le c_1 \le 3$ and $0 \le c_2 \le 3$, respectively.

The proposed method uses the equation (11) in order to properly separate H from B by adjusting c_1 and c_2. For example, if we consider all the color channels in the both color spaces, then c_1 and c_2 are 3. This indicates that all the color channels of pixel i satisfy $D_i(x) > Th_i(x)$. Or if we only take two color channels into account, c_1 and c_2 are 2. In the case of considering the characteristics of each color space, we could determine c_1 and/or c_2 individually.

As shown in Fig. 1, it represents the vector difference between μ_i at pixel i in the reference image and I_i at pixel i in the current image. Th_i is the function of the standard deviation at pixel i. It is shown how the proposed method is used to classify pixel i in each color space.



(a) (b)

Figure 1. Color Spaces and the classification of its pixel. (a) RGB Color Space (b) Normalized RGB Color Space. μ_i is the mean of pixel i and I_i is the color value of pixel i. Th_i is the threshold value at pixel i.

2.3 Background Subtraction Algorithm

The proposed method has two stages [3][4][5]. One is training background and the other is subtracting from the trained background. However, as shown in Fig. 2, each stage has two steps in the proposed method. In the first stage, we train background images and make the reference image in RGB and normalized RGB color space, respectively. Then in the second stage, we do subtract the current image from the reference image in each color space. In training background stage, we model background using equation (1). Then we determine the threshold at pixel i through equation (6) and (7). After background modeling is done in each color space, we separate the user with cast shadows from the background scene in RGB color space using equation (8). Then we quantize the result image as a binary map. As shown in Fig. 2, the created binary map is used as a mask image in normalized RGB color space. When we apply the mask image into the reference image and current image in normalized RGB color space at the same time, we simply discard cast shadows from the user because shadows have only effects on luminance. Through these two stages, we easily achieves the user image (H) without cast shadows.

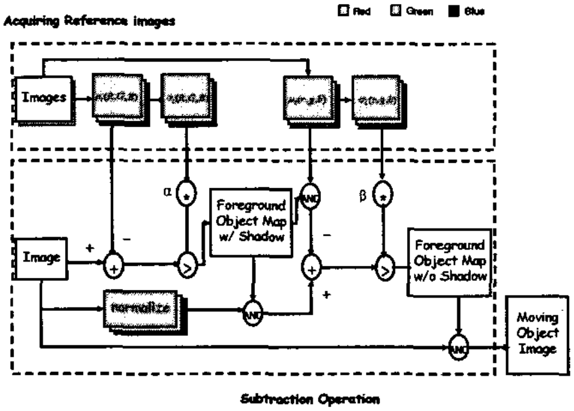
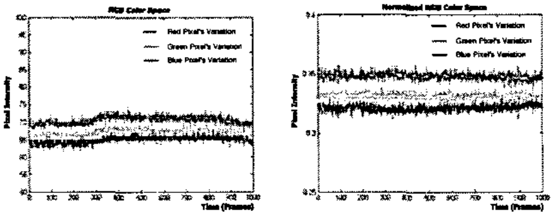


Figure 2. The proposed background subtraction algorithm. (R, G, B) and (r, g, b) denotes RGB and normalized RGB color. μ_i and σ_i represent the mean and the standard deviation at pixel i.

3. Experimental Results

As shown in Fig. 3, it shows the variation of pixel i over time in each color channel in the RGB and normalized RGB color space. The variation of pixel i over time is different in each color channel.



(a) (b)

Figure 3. The variation of pixel i over time in each color space. (a) RGB color space: Red σ=1.1436, Green σ=0.9665.40, Blue σ=0.9734 (b) Normalized RGB color space: Red σ=0.0031, Green σ=0.0025, Blue σ=0.0030

Thus, in order to subtract a current image from the reference image, we have to exploit the characteristics of pixel i individually in each color channel.

As shown in Fig. 4, it illustrates the subtraction results in the RGB and normalized RGB color space, respectively.

The results show that there are many subtraction errors to discriminate the pixel i of the user in the current image from the background scene when we use only one color space. In RGB color space, the proposed algorithm subtracts not only the user but also shadows from the background scene. In normalized RGB color space, it removes most of cast shadows around the user. However, it also removes the actual parts of the user. As shown in Fig. 4 (d), it exploits the two well-known color spaces and discriminates the user from the background scene. Although the proposed method uses the two color spaces, the result still shows many errors due to misclassifications of the pixels in RGB color space and/or normalized RGB color space.

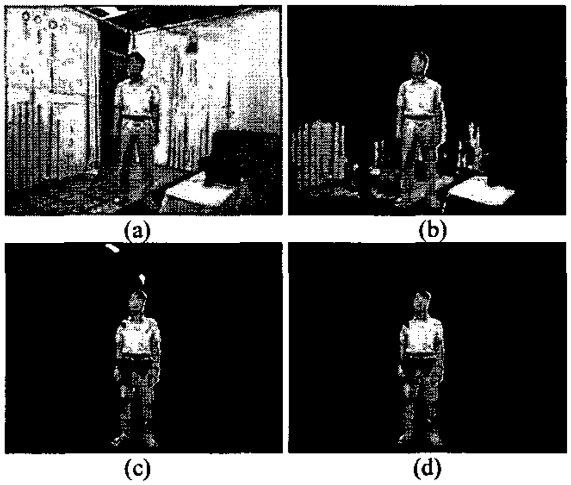


Figure 4. The subtraction results in RGB color space, in normalized RGB color space and in both color spaces. (a) A current image. (b) The subtracted image in RGB color space. (c) The subtracted image in normalized RGB color space. (d) The subtracted image in the proposed algorithm ($\alpha = \beta = F_i = f_i = 3$).

The results indicate that the color differences between the user and the background scene in RGB color space are based on either chromaticity or luminance. However, the color differences between the user and the background scene in normalized RGB color space are based on only chromaticity. In RGB color space, we differentiate chromaticity as well as luminance of the user from the background. Thus, we can observe the shadows around the segmented user. In normalized RGB color space, we differentiate only the chromaticity of the user from the background scene. We cannot see cast shadows around the user, but we can observe many false detections in the user. As shown Fig. 4 (d), the result of the proposed method discriminated shadows from the user, which uses two color spaces at the same time. However, in this stage, we just used the default determinant constants α =3 and β =3 as well as the default determinant functions F_i =3 and f_i = 3. We yet need to adjust the determinant constants empirically.

As shown in Fig. 5, it shows the results of the different determinant constants α and β which determine the threshold values in each color space. And then, it shows the result of the proposed method over the different determinant constants α and β , in which we used the default determinant functions F_i = 3 and f_i = 3.

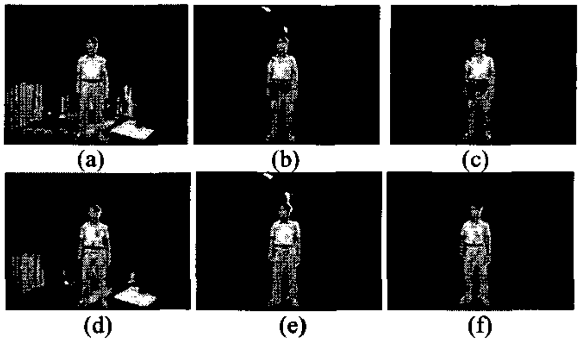
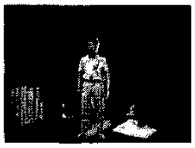
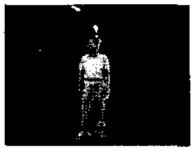
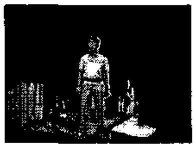
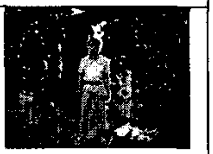
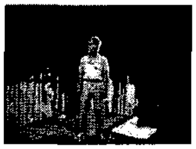
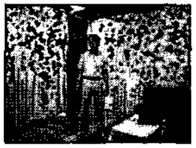


Figure 5. The determinant constants in RGB color space, in normalized RGB color space and the results. (a) α = 3 (b) β = 3 (c) the subtracted result (d) α = 4.48 (e) β = 1.06 (f) the subtracted result. The determinant functions are F_i = 3 and f_i = 3.

From the results, α = 4.48 and β = 1.06 are the optimal determinant constants for the threshold functions at pixel i during the experiments where we considered all the color channels. By the determinant constants, the results show the improvements of the subtraction in the comparison with the results of the above figures. In RGB color space, as shown Fig. 5 (a) and (b), the cast shadows around the user are reduced as the determinant constant α is increasing. In normalized RGB color space, as shown Fig. 5 (b) and (e), the misclassified of the pixels in the user are declined as the determinant constant β is increasing. As the results are shown, the proposed method is not affected by either RGB color space or normalized RGB color space, but affected by the both color spaces. Therefore, it is necessary to find the optimal determinant constants in both RGB color space and normalized RGB color space either manually or automatically for the threshold values at pixel i. However, in spite of the enhancement in the result of the subtraction, we can still improve the subtracted image of the user through the determinant functions.

After we found the optimal determinant constants for the both color space, we need to find the determinant functions in the both color space. In the determinant functions, they represent how many color channels are considered in the propose algorithm. As shown in Table 1, it shows the results through the determinant functions F_i and f_i which characterize pixel i in the images. In this experiment, we adjusted the determinant functions F_i and f_i meanwhile we fixed the determinant constants as α = 4.48 and β = 1.06.

Table 1. The determinant functions in RGB color space, normalized RGB color space and the results

Determinant Functions	RGB	Normalized RGB
$F_i = f_i = 3$		
$F_i = f_i = 2$		
$F_i = f_i = 1$		

As we expected from the results, it is the optimal results when the both determinant functions are F_i = 3 and f_i = 3. However, it has some subtraction errors in the user when the determinant function is f_i = 3 in normalized RGB color space. This leads to a false subtraction in the result image. Therefore, we choose the determinant functions are F_i = 3 and f_i = 2 rather than F_i = 3 and f_i = 3.

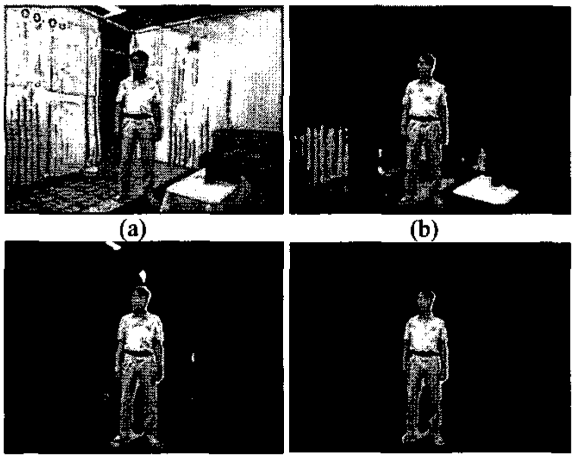


Figure 6. The subtraction results. (a) a current image (b) the determinant constant α = 4.48 and function F_i = 3 in RGB color space (c) the determinant constant β = 2.45 and function f_i = 2 in normalized RGB color space. (d) The subtracted result.

According to the experimental results, we can achieve the optimal subtracted image of the user by determining α = 4.48, β = 2.45, F_i = 3 and f_i = 2. The result clearly shows that it subtracts the user from the background scene without introducing a complex color model. However, we have to select the determinant constants and functions manually.

4. Discussion

In this paper, we proposed a pragmatic background subtraction technique for the vision-based user interface. Instead of introducing a complicated color model, we exploited the characteristics of the two well-known color spaces. The proposed method shows not only how the threshold values are chosen, bus also how to use the chosen values in the subtraction operations. We showed that the proposed determinant functions (F_i , f_i) well classified pixel i as either background or the user of interest through the experimental results. However, we need further experiments on how to find the optimal determinant constants and functions automatically.

References

[1] W. Woo, N. Kim, K. Wong and M. Tadenuma, "Sketch on Dynamic Gesture Tracking and Analysis Exploiting Vision-based 3D Interface," in Proc. SPIE PW-EI-VCIP'01, vol. 4310, pp. 656-666, Jan. 2001

[2] Kida, K., Ihara, M., Shiwa, S., Ishibashi, S., "Motion tracking method for the CAVETM system", Signal Processing Proceedings, 2000 WCCC-ICSP 2000. 5th International Conference on , 859 -862 vol.2, 2000

[3] T. Horprasert, D. Harwood, and L.S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection,"Proc. IEEE ICCV'99 FRAME-RATE Workshop, Kerkyra, Greece, September 1999

[4] Ahmed Elgammal, David Harwood, and Larry Davis, "Non-parametric Model for Background Subtraction," 6th European Conference on Computer Vision, Dublin, Ireland, June/July 2000.

[5] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis "Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance", Proceedings of the IEEE, July 2002.

[6] C. Kim, W. Woo, and H. Jeong, "Determination of Optical Flow by Stochastic Model," Journal of the Korea Information Science Society (KISS), vol.19, no.6, pp.581-594, Nov., 1992.

[7] W. Woo and H. Jeong, "Stochastic Model for Unification of Stereo Vision and Image Restoration," Journal of the Korean Institute of Telemetric and Electronics (KITE), vol.29-B, no.9, pp.37-49, Sep., 1992

[8] W. Woo, N. Kim and Y. Iwadate, "Object Segmentation for Z-keying Using Stereo Images," in Proc. IEEE WCC-ICSP'00, vol.2, pp.1249-1254, Aug. 2000.