Real-time Color Correction for Marker-based Augmented Reality Applications*

Wonwoo Lee, Woontack Woo
GIST U-VR Lab.
Gwangju, 500-712, S. Korea
{wlee, wwoo}@gist.ac.kr

Abstract

Exploiting color information in an augmented reality (AR) application running on a mobile device becomes problematic due to the unknown lighting condition of the environment. In this paper, we propose a real-time color correction method for AR applications that use markers with color patterns. We identify the regions of color patterns in a marker image and estimate the black and white color points using the marker image itself. Then, the pixel colors in the pattern regions are corrected using the scaling vector. The proposed color correction method requires no assumptions on lighting conditions and is fast enough to run in real-time on a ultra mobile PC platform. We demonstrate the performance of our method with the real world data sets.

1. Introduction

In line with the advances in mobile device platforms, such as mobile phones and ultra mobile PCs, many AR applications have been developed on them [10]. In many mobile AR applications, fiducial markers have been frequently used due to their robustness and ease-of-use. There exist many planar marker systems, such as Data Matrix[4], QR code[9], ARTag[5], and ARToolkit[6], etc. The type of coded markers [4, 9] are useful to embed information, but not suitable for AR applications due to the high resolution of the patterns. The ARToolkit marker [6] is one of the most popular marker systems, since it is simple and robust enough to be used in AR applications. ARTag provides better robustness in tracking and recognition than ARToolkit. It exploits digitized patterns to recognize the pattern inside the border.

While most fiducial marker systems consist of black patterns printed on white background, there have been approaches that exploit color information as an important cue [3, 7]. By adopting colors, it is possible to increase the amount of information represented by a marker. To exploit color information, it is required to distinguish different colors robustly. However, identifying colors in a mobile AR environment becomes problematic due to the low quality of the images captured by a built-in camera In addition, the colors of illuminants incident on markers and white balance setting of the camera make imaged colors from their original colors. Thus, color identification becomes difficult in mobile AR environment.

Many computational color constancy algorithms have been proposed to recover the colors of a scene precisely [1, 2]. However, the computational color constancy algorithms are not desirable in AR applications due to the following reasons: 1) the mobile devices have limited computing performance and thus the color correction should not take computational cost; 2) what we are interested in is only the markers' areas in the image while computational color consistency algorithms perform color correction on the entire image. Julien *et al.* proposed an on-line photometric calibration approach for AR [8]. However, their approach is only for static cameras, not for moving cameras.

In this paper, we propose a real-time color correction method for AR applications that use markers with color patterns. First, we identify the regions of color patterns in a marker image. Then, we estimate the black and white color points using the marker image itself. A scaling vector that transforms one color vector under the current illuminant to another under a canonical illuminant is found from the estimated black and white points. Finally, the pixel colors in the pattern regions are corrected using the scaling vector through component stretch. The advantages of the proposed methods are as follows: 1) the proposed color correction method does not require any assumptions on lighting conditions or illuminations of the environment since the sample colors for correction is obtained from the image; 2) the color correction is applied to each marker independently, and thus our method can conduct color correction robustly even though two color markers are under different

*This research was supported in part by the CTI development project of KOCCA, MCST in S. Korea and in part by the basic research project through a grant provided by the Gwangju Institute of Science & Technology in 2008.

Bruce Thomas et al. (Eds.): IWUVR 2009, pp. 32-35, 2009.
ISBN 978-0-9806098-0-6

illuminants in the same image; 3) we perform color correction only on the regions of the color patterns and thus, our method is fast enough to run on mobile devices.

This paper is organized as follows. We describe the details of our color correction approach in Section 2. Experimental results are shown and evaluated in Section 3. We provide conclusions and future works in Section 4.

2. Our approach

2.1. Pattern region identification

The input image of a color marker is firstly converted to a gray image and binarized. For binary image generation a simple thresholding method or adaptive thresholding method is usually used. These two thresholding methods show good results with the black and white markers and have been adopted in many AR systems [5, 6]. When the marker has color patterns instead of black patterns, both of thresholding methods generally do not provide promising results since some of color pattern regions are lost after binarization.

To robustly identify the pattern regions in a marker image, we perform per-channel binarization in each of R, G and B channel images, and merge the binarization results into one binary image. We apply the adaptive thresholding method to each channel image and the binary images of all three channels are merged through AND operation. This per-channel binarization is simple, but effective for binarization of color marker images.

2.2. Color correction

To estimate the illumination incident on a marker, we focus on the fact that the color marker has both black and white areas by itself. The square border is black and the region inside the border is white except the color patterns. We estimate both color points in RGB space by using the corresponding regions in the marker image. Then, the component stretch is applied to pixels in the regions of color patterns. The color correction task is independently conducted with every marker region. What we estimate is not the global illumination of a scene but the local illumination incident on a marker. Thus, it is possible to perform color correction on two markers although they are under different illuminations in an image.

Firstly, we find the pixels of the black border region $\Re_b$ and the white region $\Re_w$ inside the border based on the connected component analysis. To reduce the errors on the boundary of the regions, we perform the erosion on the detected regions. After the black and the white regions are identified, we estimate the black point $\mathbf{B}^e\,(b^e_R, b^e_G, b^e_B)$ and

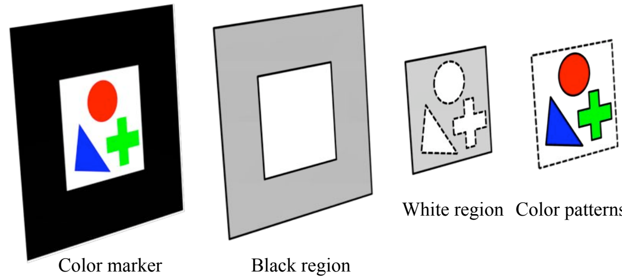

Figure 1. Composition of a marker. Black and white regions are used to estimate B^e and $\mathbf{W}^e$.

white point $\mathbf{W}^e\,(w^e_R, w^e_G, w^e_B)$ under the unknown illumination. Finally, the R, G, and B channels are stretched independently by using $\mathbf{B}^e$ and $\mathbf{W}^e$. We adopt the diagonal model of illumination change[2], through which the image taken under one illuminant can be transformed to the image taken under another illuminant by scaling each channel independently.

$\mathbf{B}^e$ and $\mathbf{W}^e$ are estimated from the samples obtained from $\Re_b$ and $\Re_w$. For color modeling, we assume that the pixel colors follow the Gaussian distribution in the region $\Re_b$ and $\Re_w$. We build Gaussian models $N_b\,(\mathbf{m}_b, \sigma_b)$ and $N_w\,(\mathbf{m}_w, \sigma_w)$ with respect to $\Re_b$ and $\Re_w$. We define $\mathbf{B}^e$ and $\mathbf{W}^e$ as

$$\mathbf{B}^e = \mathbf{m}_b + \Gamma \cdot \sigma_b \tag{1}$$

$$\mathbf{W}^e = \mathbf{m}_w + \Gamma \cdot \sigma_w \tag{2}$$

where the Γ represents a scaling vector to control $\mathbf{B}^e$ and $\mathbf{W}^e$ based on the standard deviation vector.

Thus, finding $\Gamma\,(\gamma_R, \gamma_G, \gamma_B)$ that minimizes a cost of the mapping $\mathbf{B}^e$ to $(0, 0, 0)$ and $\mathbf{W}^e$ to $(255, 255, 255)$ results in optimal $\mathbf{B}^e$ and $\mathbf{W}^e$ for color correction. Since the color correction is conducted by scaling each channels independently, γ_R, γ_G, and γ_B are estimated with the R, G, and B channel, respectively.

Considering a single channel, let I be the intensity of a pixel in the channel and I' be the intensity after the color correction. Ideally, I' should be the same as a desired intensity I'_d after the color correction. Since there exist noises on the pixel colors, $I_\prime$ is not identical to I_d. Thus, the cost of mapping from I to I' is defined as the squared difference between I'_d and I'.

$$C(I') = \|I_d - I'\|^2 \tag{3}$$

The desired pixel intensity is 0 for the black region $\Re_b$ and 255 for the white region $\Re_w$. The cost of mapping all the pixels of the region $\Re_b$ and $\Re_w$ is defined as Equation 4 and 5, respectively.

$$E_b = \frac{1}{N_b}\sum_{i=1}^{N_b} C\left(I_{bi}\right) \tag{4}$$

$$E_w = \frac{1}{N_w}\sum_{j=1}^{N_w} C\left(I_{wj}\right) \tag{5}$$

where N_b and N_w are the number of pixels in the regions $\Re_b$ and $\Re_w$. I_{bi} and I_{wj} refer to the intensities of pixels in the regions.

The total cost of the mapping is defined as the sum of E_b and E_w. By computing the costs for all the channels and minimizing them, we estimate γ_R, γ_G, and γ_B and obtain the corresponding black point $\mathbf{B}^e$ and white point $\mathbf{W}^e$.

$$E_{total} = E_b + E_w \tag{6}$$

After the black and white points are estimated, we perform color correction with the pixels of the color patterns. For a pixel $P\left(I_R, I_G, I_B\right)$ in a pattern, the corrected pixel color $P'\left(I'_R, I'_G, I'_B\right)$ is computed from the estimated $\mathbf{B}^e$ and $\mathbf{W}^e$ from Equation 7.

$$\begin{aligned} I'_R &= 255\left(\frac{I_R - b^e_R}{w^e_R - b^e_R}\right) \\ I'_G &= 255\left(\frac{I_G - b^e_G}{w^e_G - b^e_G}\right) \\ I'_B &= 255\left(\frac{I_B - b^e_B}{w^e_B - b^e_B}\right) \end{aligned} \tag{7}$$

Since color correction task is performed on every marker independently, the local color correction enables us to perform robust color correction with the markers each of which has different illumination conditions. When we use several markers in a marker-based AR application, for instance, each marker can be under different illumination depending on the lighting condition. In this case, a preset white balance may successfully work with some of the markers, but may not provide good color correction results to the others.

3. Experimental Results

We implemented the proposed color correction method on a UMPC platform that has 1GHz CPU. We performed experiments with ARToolkit style markers that contain three color patterns of R, G and B. The color correction speed is up to 18 frames/second with 320x240 video captured from a USB camera. The performance varies depending on the size of a marker in the video, but it maintains real-time performance.

Figure 2 shows the results of color correction with respect to the images captured under different illuminations[1].

[1] Note that the color correction results shown in the figures may not be identifiable if this document is printed in gray-scale.

Figure 2. Color correction results under the different illuminants: all markers are under the same illumination(top); two markers are under different illuminations (bottom)

In the top figure, the colors of markers' patterns vary according to the illuminations before color correction. The images at the top row are more reddish, bluish, and greenish colors. After the color correction, the black color of the marker border and the white color of the regions inside the border are much improved and they are more like the real black and the white. The colors of the patterns inside the border became closer to their original color. In the bottom of Figure 2, two markers are under different lighting condition in the same image, and our local correction approach successfully performed color correction on each marker. According to our experiments the color correction showed good results and real-time performance.

The chromaticities of a blue pattern before and after color correction are shown in Figure 3. We captured four image sequences, where the camera moves freely around a marker, under four different illuminations. As we can see in Figure 3, the same color points under different illuminations are scattered before color correction. These color variations make it difficult to perform color identification robustly. After the color correction is conducted (bottom), the colors of a pattern under different illuminations became more similar to one another. Thus, performing the color segmentation under different illuminations is possible with the proposed color correction method.

Table 1 shows the statistics of corrected colors, which

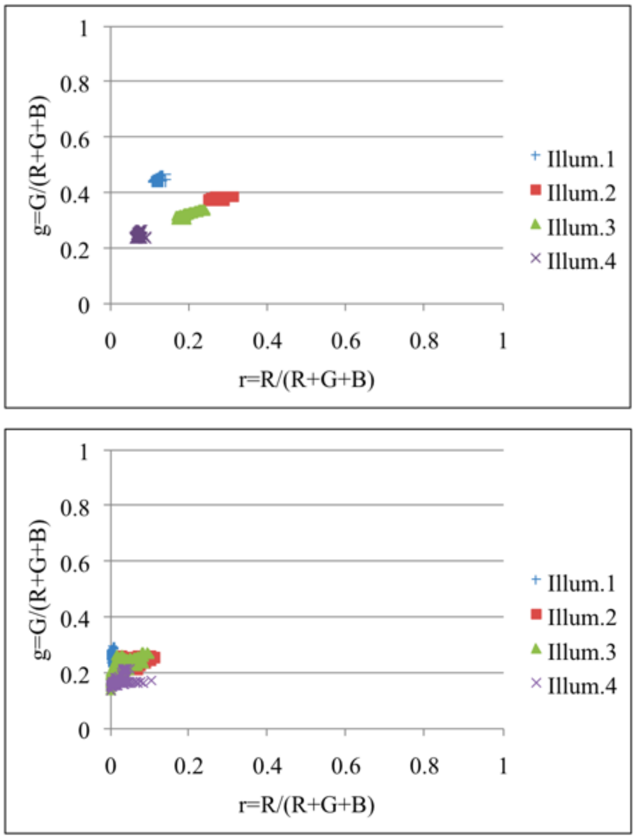

Figure 3. Chromaticities under different illuminations: before color correction (top; after color correction (bottom)

shows how much the corrected colors vary. We computed the mean and standard deviation of the corrected colors with respect to each channel in normalized color space. As we can see, the corrected colors do not vary much although they are obtained under different illuminations. Based on this result, it is possible to determine a threshold value from the statistics, e.g. by scaling the standard deviation in each channel since the purpose of color correction is distinguishing different colors.

Table 1. Statistics of the corrected colors.

Patterns	Red		Green		Blue	
	Mean	STD	Mean	STD	Mean	STD
Ch_R	0.87	0.015	0.34	0.02	0.03	0.02
Ch_G	0.1	0.017	0.48	0.02	0.22	0.03
Ch_B	0.02	0.01	0.18	0.02	0.75	0.05

4. Conclusions

In this paper, we proposed a real-time color correction method for mobile AR applications. It automatically estimates the black and white points without any prior knowledges on illuminations. Our method can perform color correction on markers, each of which is under different illuminations since the local illumination incident on each color marker is estimated. According to our experiments the color correction showed good results and real-time performance. As future works, the experiments with off-the-shelf mobile phones will be conducted since they emerge as a major platform of mobile AR applications.

References

[1] K. Barnard, V. C. Cardei, and B. V. Funt. A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–984, 2002.

[2] K. Barnard, L. Martin, A. Coath, and B. Funt. A comparison of computational color constancy algorithms. ii. experiments with image data. *Image Processing, IEEE Transactions on*, 11(9):985–996, Sep 2002.

[3] Colorcode. *http://www.colorzip.com/*.

[4] DataMatrix. *International Organization for Standardization, Information technology: Automatic identification and data capture techniques - Bar code symbology - Data Matrix (ISO/IEC 16022:2000)*, 1st edition, 2000.

[5] M. Fiala. Artag, a fiducial marker system using digital techniques. In *Proc. of CVPR '05*, pages 590–596, 2005.

[6] H. Kato and M. Billinghurst. Marker and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of IWAR '99*, page 85, 1999.

[7] W. Lee and W. Woo. Exploiting context-awareness in augmented reality applications. In *International Symposium on Ubiquitous Virtual Reality*, pages 51–54, July 2008.

[8] J. Pilet, A. Geiger, P. Lagger, V. Lepetit, and P. Fua. An all-in-one solution to geometric and photometric calibration. In *International Symposium on Mixed and Augmented Reality*, pages 69–78, October 2006.

[9] QRCode. *International Organization for Standardization, Information technology: Automatic identification and data capture techniques - Bar code symbology - QR Code (ISO/IEC 18004:2000)*, 1st edition, 2000.

[10] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Pervasive 2005*, volume 3468 of *LNCS*, pages 208–219. Springer, 2005.