A Real-time Color-matching Method Based on SmartPhones For Color-blind People

Myoungbeom Chung, Hyunseung Choo
College of Information and Communication Engineering
Sungkyunkwan University
Suwon, Republic of Korea
e-mail: nzin@ssu.ac.kr, choo@skku.edu

***Abstract*—In this paper, we propose a real-time color-matching method based on smart phones for color-blind people. Most people have no trouble seeing color, but about 8% of males and less than 1% of females have faulty color perception from birth. Recently, some applications based on mobile phones have become available for color-blind people. However, most cannot compare colors in real time but just provide color values and names based on the capture images of mobile cameras. However, the proposed method can be used to match colors in real time, as it can report similar colors based on comparisons between capture images and real-time camera images on smart phones. To evaluate the efficacy of the proposed method, we conducted a color-matching experiment, and the matching success rate was 99%. Therefore, the proposed method will be a useful technique for color-blind people.**

Keywords-color-matching application; smartphone; real-time color matching; color-blind people.

I. INTRODUCTION

Color is very important in our lives. Different aspects of color, such as its warmth, coldness, or softness, make different impressions on us. Combining various colors can change our impressions of and feelings about things. Most people have no trouble seeing color, but about 8% of males and less than 1% of females have faulty color perception from birth. Although this is not a serious issue, it can be inconvenient.

To solve the problems related to color blindness, Tomoyuki proposed real color expression technology based on color correction using a Charge-Coupled Device (CCD), a computer, and a Head-Mounted Display (HMD) [1][2]. Koji devised a color information technique whereby the color value of a specific location shows on a color wheel using the camera feature of a phone [3]. Simon proposed a color confirmation application, simulating a deutan defect and identifying color using the color wheel from the camera image capture function of a smart phone [4]. In their recent research, Manaf and Harwahyu traced the finger-pointing position from a real-time camera image on a smartphone and determined the color value of the pointing location through speech using augmented reality [5][6]. However, there are some problems with the existing studies. They show general color value or determine colors based only on an analysis of the input image. They cannot identify undefined colors at the existing method or match colors. For example, if a color-blind user employs the existing color-matching method to find a matching sock in a pile of variously colored socks, he has to take many photos to match all the socks and must remember many names of colors in order to find the same color socks. Moreover, the existing application, which applies augmented reality using finger detection on a smart phone, is uncomfortable because one hand has to hold the smart phone and the other hand has to point out something.

To solve the problems associated with the existing method, we propose a real-time color-matching method based on smart phones for color-blind people. In the proposed method, takes a photo to finding color using camera of smart phone; the photo is located on the left side of the smart phone screen. The real-time input image from the smart phone camera is shown on the right side of the screen, and an analysis of the real-time image is conducted. Therefore, the method can be used to easily match the same color in real time because the user can see the capture image and the real-time input image on the same screen at the same time. The proposed method uses Red-Green-Blue (RGB) color value and the Hue–Saturation–Intensity (HSI) color model for color matching. The method also uses a fixed center range of the input image to rapidly process color analysis of the real-time image. For a color comparison, the user can look at the capture image on the left side of the screen and the real-time input image on the right side of the screen and move the match range using a pointer. The similarity of the pointed range of the capture image and the center range of the real-time image are compared using cosine similarity, and the user is notified if the colors match. Thus, the proposed method can be very useful for color comparisons. To evaluate the performance of the proposed method, we created a real-time color matching application based on the iOS mobile operating system. We conducted a color-matching test using a printed color image and the application and achieved a 99% success rate. Therefore, the proposed method would be a useful technique for color-blind people to match colors.

This paper is organized as follows. In section 2, we explain the conversion method used to change the RGB of an image to HSI color space and the color match method, which uses cosine similarity. In section 3, we explain the screen

 ISBN: 978-1-61208-353-7

composition of the proposed application and the color-matching algorithm that supports color information for color-blind people. In section 4, we report on the color matching test results of the proposed application. Section 5 is the conclusion.

II. RELATED WORK

In this section, we explain the conversion method used to change the RGB of an image to HSI color space to match color values and the color value matching method, which uses cosine similarity. The RGB value of an image ranges from 0 to 255, according to each color. The color value of one pixel is determined based on the combination of colors in (1).

$$Color\ value\ of\ one\ pixel = R + G + B \quad (1)$$

In (1), **R** is red, **G** is green, and **B** is blue. The color value of one pixel on a computer or smart phone uses a specific amount of memory, as seen in Fig. 1, the 24 bit format of RGB color [7].

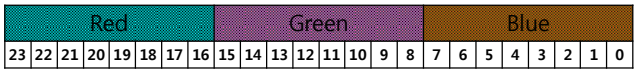


Figure 1. Memory use of smart-phone to show RGB color.

In Fig. 1, Red, Green, and Blue each have 8 bits. The range of Red is from 16 to 23, the range of Green is from 8 to 15, and the range of Blue is from 0 to 7. Thus, the RGB color of one pixel is calculated as in (2).

$$RGB\ Color = (R \times 65536) + (G \times 256) + B \quad (2)$$

In (2), Red memory space is multiplied red value by 65536 (256×256) for the 16–23 position, and Green memory space is multiplied green value by 256 for the 8–15 position. These values can be shown as 16 digit notations, according to each color. The RGB color calculated from the image can be changed to HSI color space according to (3) [8][9].

$$H = \cos^{-1}\left[0.5 \times \frac{\{(R-G)+(R-B)\}}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right] \quad (3)$$
$$S = \frac{1-\min(R,G,B)}{I}, I = \frac{(R+G+B)}{3}$$

The reason the RGB color value is changed to HSI color space is that we want to compare each color using only the hue value of each pixel. Hue of HSI color space is location of color at the visible domain of electromagnetic spectrum of light and it describes the color itself in the form of an angle between [0, 360] degree in Fig. 2. 0 degree mean red, 120 means green 240 means blue. 60 degree is yellow, 300 degrees is magenta. The saturation component signals how much the color is polluted with white color. The range of the S component is [0, 1]. And the intensity range is between [0, 1] and 0 means black, 1 means white. We can divide the HSI value by the RGB color value and can compare that with each pixel, according to hue value.

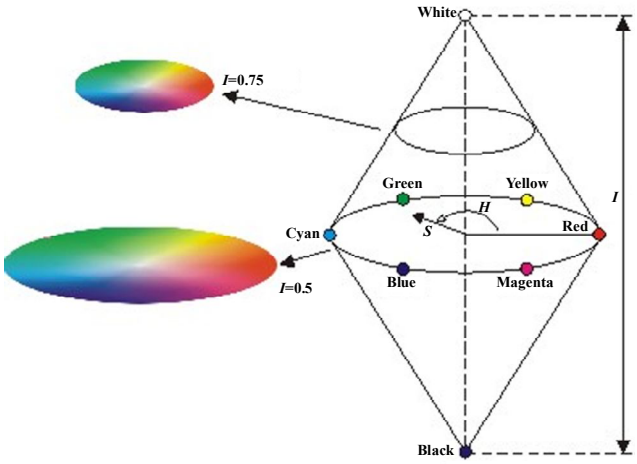


Figure 2. How the HSI color space represents colors

The color-matching method using cosine similarity is a distance calculation method that considers the direction of the color value of each pixel that needs to be compared. Normally, we use Euclidean distance to calculate the distance of comparison items. However, this method does not consider the direction of items [10]. On the other hand, the cosine similarity method considers the direction of each item and is often used in the item search field, in item similarity measurement, in vector space models, etc. Equation (4) is the calculation method for cosine similarity.

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\left\|\vec{i}\right\|\left\|\vec{j}\right\|} = \frac{\sum_{k=1}^{n} i_k \cdot j_k}{\sum_{k=1}^{n} i_k^2 \sum_{k=1}^{n} j_k^2} \quad (4)$$

In (4), ***sim(i, j)*** is the similarity value of items ***i*** and ***j***. $\vec{i}$ and $\vec{j}$ are the vector values of each item. This is useful when finding the similarity between two items. A perfect correlation will have a score of 1 (or an angle of 0) and no correlation will have a score of 0 (or an angle of 90 degrees). According to (4), if the value of ***sim(i, j)*** is lower, the similarity of each item is bigger. On the contrary, if the value of ***sim(i, j)*** is higher, the similarity is smaller.

III. THE PROPOSED APPLICATION AND REAL-TIME COLOR-MATCHING ALGORITHM

In this section, we explain the screen composition of the proposed application and the color- matching algorithm that supports color information for color-blind people in real time. The screen composition of the proposed application is as in Fig. 3. In Fig. 3, the capture image for color matching is located on the left side of the smart phone screen, and the

 ISBN: 978-1-61208-353-7

real-time input image from the smart phone camera is located on the right side. The user can select the capture image on the left side from the images in the photo library or can display a capture image selected from the photos taken using the smart phone camera.

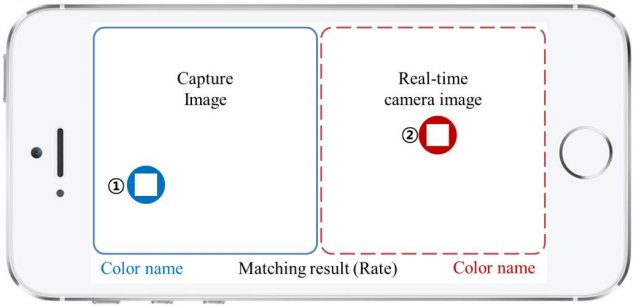


Figure 3. Screen composition of the proposed application.

The point icon (① in Fig. 3) of the capture image can be moved with a touch of the user when he/she wants to change the color matching position. The real-time camera image on the right side is the inputted image from the smart phone camera, and the center point icon (② in Fig. 3) is the color-matching range in real time. However, this icon cannot be moved by the user's touch. The pixel range for color-value matching is N×N, and each pixel range on the left and right sides calculates the RGB color value and changes it to HSI color space. The RGB color value of each pixel is stored as an array type, as in (5).

$$LR\,[a(n-1)+b] = Red\ (a, b)\ of\ capture\ image$$

$$RR[a(n-1)+b] = Red\ (a, b)\ of\ real\text{-}time\ image \quad (5)$$

In (5), $\boldsymbol{a}$ and $\boldsymbol{b}$ are the $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates of the related pixel, and n is the row number of the selected range. $\boldsymbol{L_R}$ and $\boldsymbol{R_R}$ are the red value arrays for the pixel range of the left and right images, and $\boldsymbol{L_G}$, $\boldsymbol{R_G}$, $\boldsymbol{L_B}$, and $\boldsymbol{R_B}$ are stored as in (5). The HSI color value, which is changed by (3), is stored as an array type, as in (6); we use this in the color-matching algorithm using cosine similarity.

$$LH[a(n-1)+b] = Hue\ (a, b)\ of\ capture\ image$$

$$RH[a(n-1)+b] = Hue\ (a, b)\ of\ real\text{-}time\ image \quad (6)$$

In (6), $\boldsymbol{L_H}$ and $\boldsymbol{R_H}$ are the hue value arrays for the pixel range, and $\boldsymbol{L_S}$, $\boldsymbol{R_S}$, $\boldsymbol{L_I}$, and $\boldsymbol{R_S}$ are stored as in (6).

The color-matching algorithm that supports color information in real time is shown in Fig. 4. The algorithm calculates the color-matching value using cosine similarity with each RGB and hue color value. It returns "Excellent", "Good", or "Bad" as the matching result, according to similar rates between each color. It also provides familiar color names for the user from the HSI value table using HSI color values. In Fig. 4, to obtain color-matching values, the proposed algorithm uses two types of cosine similarity. One is a similar rate using the average value of each color array; we call it the "average similar rate". The other is a similar rate using the average value of the similar rate between each pixel; we call it the "each pixel similar rate". Thus, the color-matching value is determined using the average similar rate and the each pixel similar rate, according to the pseudocode in Fig. 5.

```
If (Average similar rate is over T1) Then
  If (Each pixel similar rate is over T1) Then
    Return Excellent;
  Else If (Each pixel similar rate is over T2) Then
    Return Good;
  End If
Else If (Average similar rate is over T2) Then
  If (Each pixel similar rate is over T1) Then
    Return Good;
  END If
Else
Return Bad;
```

Figure 5. Pseudocode for color-matching results

In Fig. 5, $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ are each threshold values for color-matching results. The pseudocode returns the correct match when both the average similar rate and the each pixel similar rate is over $\boldsymbol{T_2}$. If the average similar rate or the each pixel similar rate is not over $\boldsymbol{T_2}$, the matching result is "Bad", and the color on the left side does not match that on the right side. Then, the color name is given to the user as supporting

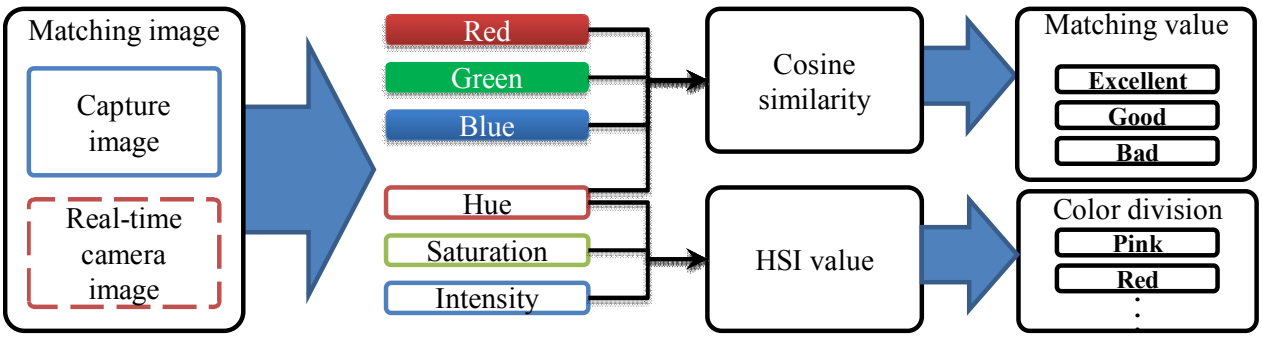


Figure 4. The flow of the color-matching algorithm

information use each array value of HSI. The color is returned one color of 16 kinds color name according to the color names defined by Harwahyu [6].

IV. EXPERIMENTS AND EVALUATION

In this section, we introduce the real-time color-matching application based on smart phones for color-blind people and explain the experiments and results to evaluate the performance of the proposed method. The application can work on iOS 6; we created it using Xcode 5. We used iPhone 5 of Apple Inc. as test device and screen size of iPhone 5 is 4 inch widescreen. And then, we set the brightness level of screen to maximum. To analyze the input image from the smart phone camera in real time, we used the AVCaptureVideoDataOutputSampleBufferDelegate protocol, which is supported by Apple and has two methods [11]. This protocol works on background of application and can converts endless captured big image of the smart phone camera to small input image. Figure 6 shows the captured image for the developed application, according to the proposed method.

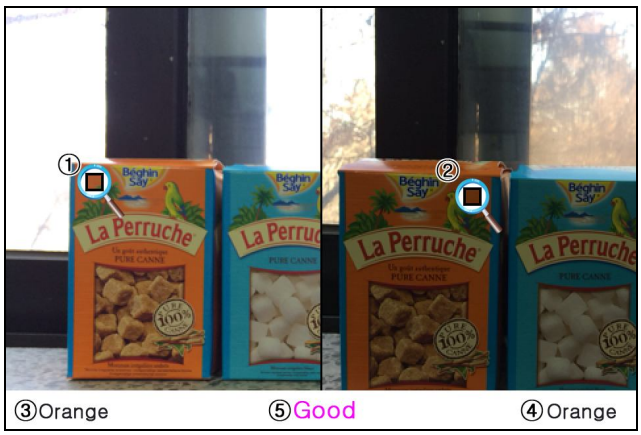


Figure 6. Capture of real-time color-matching application

① in Fig. 6 is a point icon that can be moved by user touch for color matching, and ② in Fig. 6 is the center pixel range of the real-time input image from the smart phone camera. ③ and ④ in Fig. 6 are the color name results of the pixel range of each image according to the defined color name, and ⑤ in Fig. 6 shows the color-matching results of the pixel range for both sides.

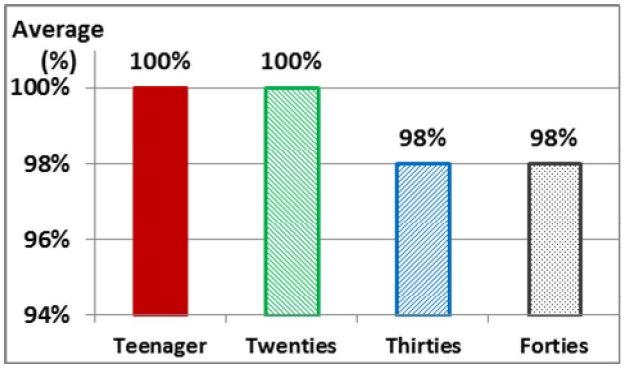
Next, we evaluated the performance of the proposed method. Twenty participants, five in each of four age groups (teenagers, twenties, thirties, and forties) were included in the experiment. To prevent the participants seeing the colors, we used the simulation eyeglasses for achromatopsia test that Tomoyuki used. The color-matching target used in evaluating the performance is a 3×3 table consisting of nine random defined color names. After participants take a capture image of the 3×3 table using the application, they find a matching color in a re-ordered 3×3 table in which the color positions are changed, even though it has the same nine colors within 5 seconds for color. In this experiment, we use 20×20 pixels as the pixel range. The values of T_1 and T_2 in the color-matching result are 90% and 80%, respectively. Figure 7 is a capture image of the smart phone screen of a participant. In Fig. 7, the capture image on the left side has green, yellow, light blue, pink, dark red, light green, red, white, and magenta from the top left to the bottom right. Each participant tries to find the matching color by moving the point icon to the yellow location.

Figure 8 shows the result of the color-matching experiment using the proposed application. Figure 8(a) shows the average results according to age group, and Figure 8(b) shows the average time it tool for participants to find the color matches. In Fig. 8(a), we see that the participants in the teenager and twenties groups could find the matching colors within the time limit. However, the participants in the thirties and forties groups could not find the matching colors within the time limit. Yet, they did not find a matching color on time only, and they found matching colors after the time limit was up.

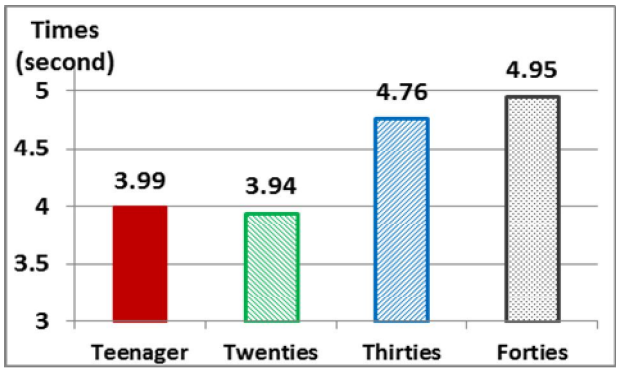
Figure 7. Screen capture of proposed application during the color-match experiment

 ISBN: 978-1-61208-353-7

Thus, the average rate for matching the colors is 99% within the time limit. If there was no time limit, it would be 100%. In Fig. 8(b), the participants in the teenager and twenties groups took about four seconds to match the colors, which is one second before the time limit was up. The participants in the thirties and forties groups took about the same amount of time as the time limit. The reason the participants in the teenager and twenties groups were faster than those in the thirties and forties groups is that they consider match color experiments like "Find the correct picture", and can concentrate better. Based on the results, the proposed method and application can successfully compare and match colors, no matter the user's age group. Therefore, the proposed method would be an effective real-time color-match method for color-blind people.



(a)



(b)

Figure 8. The result of color matching: (a)Average results according to age group, (b) Average time the method took for participants

V. CONCLUSION

The proposed method based on smart phones can be used to compare and match colors in real time, and it provides familiar color names. Thus, it is very useful for color-blind people. The proposed method had a 99% color-matching success rate over all age groups. Because it takes about 4.5 seconds for color matching, we think the proposed method could be implemented to help color-blind people.

In the future, we will determine how to improve the color similarity matching algorithm and the user interface (UI)/user experience (UX) to make the proposed method more useful and comfortable for color-blind people. We will conduct experiments with color-blind participants using the proposed application and various colors found in everyday life.

ACKNOWLEDGMENT

This research was supported in part by MSIP and MOE, Korean government, under IT R&D Program[10041244, SmartTV 2.0 Software Platform] through KEIT, Basic Science Research Program(NRF-2013R1A1A2061478) and PRCP(NRF-2010-0020210) through NRF, respectively.

REFERENCES

[1] T. Ohkubo,and K. Kobayashi, "A Color Compensation Vision System for Color-blind," In SICE Annual Conference 2008, Aug. 2008, pp. 1286-1289. doi:10.1109/SICE.2008.4654855

[2] T. Ohkubo, K. Kobayashi, K. Watanabe, Y. Kurihara, "Development of a Time-sharing-based Color-assisted Vision System for Persons with Color-vision Deficiency," In SICE Annual Conference 2010, Aug. 2010, pp. 2499-2503. ISBN:978-1-4244-7642-8

[3] V. K. Y. V. S. Kondo, and V. Y. Tsuchiya, "Development of Color-Distinguishing Application 'ColorAttendant'," FUJITSU Sci. Tech. J, vol.45, no.2, Apr. 2009, pp. 247-253.

[4] S. Schmitt, S. Stein, F. Hampe, D. Paulus, "Mobile Services Supporting Color Vision Deficiency," 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM 2012), May. 2012, pp. 1413-1420. doi:10.1109/OPTIM.2012.6231860

[5] A. S. Manaf, and R. F. Sari, "Color Recognition System with Augmented Reality Concept and Finger Interaction: Case Study for Color Blind Aid System," 9th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering 2011), Jan. 2012, pp. 118-123. doi:10.1109/ICTKE.2012.6152389

[6] R. Harwahyu, A. Sheffildi Manaf, B. Sri Ananto, B. Adi Wicaksana, R. Fitri Sari, "Implementation of Color-blind Aid System," Journal of Computer Science, vol.9, no.6, Jun. 2013, pp. 794-810. doi:10.3844/jcssp.2013.794.810

[7] RGB Color Codes Chart [Online], Available from: http://www.rapidtables.com/web/color/RGB_Color.htm. 2014. 04.10

[8] M. Chung, and I. Ko, "Intelligent Copyright Protection System using a Matching Video Retrieval Algorithm," Multimedia Tools and Applications, vol.59, no.1, Jul. 2012, pp. 383-401. doi:10.1007/s11042-011-0743-z

[9] S. M. Dominguez, T. Keaton, A. H. Sayed, "Robust Finger Tracking for Wearable Computer Interfacing," In Proceedings of the 2001 workshop on Perceptive user interfaces(ACM. 2001), Nov. 2001, pp. 1-5. doi:10.1145/971478.971516

[10] G. Qian, S. Sural, Y. Gu, S. Pramanik, "Similarity between Euclidean and Cosine Angle Distance for Nearest Neighbor Queries," In Proceedings of the 2004 ACM symposium on Applied computing (ACM. 2004), Mar. 2004, pp. 1232-1237. doi:10.1145/967900.968151

[11] Apple Inc. [Online], Available from: https://developer.apple.com/library/mac/documentation/AVFoundation/Reference/AVCaptureVideoDataOutputSampleBufferDelegate_Protocol/Reference/Reference.html. 2013.09.17

 ISBN: 978-1-61208-353-7