What Color? A Real-time Color Identification Mobile Application for Visually Impaired People

Sara A. Al-Doweesh[1], Felwah A. Al-Hamed[1], and Hend S. Al-Khalifa[2]

[1] Center of Excellence for Telecom Applications,
King Abdulaziz City for Science and Technology, Saudi Arabia
{saldoweesh,falhamed}@kacst.edu.sa
[2] Information Technology Department, College of Computer and Information Science,
King Saud University, Riyadh, Saudi Arabia
hendk@ksu.edu.sa

Abstract. In this paper we present an iPhone application that facilitates the operation of detecting colors for blind and visually impaired people in real-time. In order to detect colors, we take the input from the device camera then we process pixel values to produce the color using HSL color space, the detected color will be displayed as label on screen as well as uttering it. Moreover, the application can view a set of colors that match a specific color to help blind people choosing clothes before promenading. We tested the application on a set of blind and visually impaired people to evaluate the application accuracy and usability. Our evaluation showed that the application provides high detection accuracy of colors in different lighting conditions. Furthermore, the application satisfies its users' needs.

Keywords: Color Detection, Visually Impaired, Clothes Matching, Hue-Saturation-Luminance (HSL), Real-time Identification, Camera Phone.

1 Introduction

Latest global estimations indicate a dramatic increase in the magnitude of visual impairment in the world. Blind and visually impaired community has the right to live as normal people and we are responsible of facilitating their lives [1]. One of the challenging problems facing them is recognizing the color of things around.

Despite the existence of other techniques of color identification, such as electronic color detectors and tactile tags for marking clothes. Yet, these devices are stand-alone, which means the blind person has to carry them all the time and bear its weight. Furthermore, blind people who had no sense of color before have no way of knowing if a set of two or more color-combinations of clothes match or not [2].

According to problems mentioned previously, we decided to develop an iPhone application that detect the color of a chosen area and utter its name in Arabic. Moreover, it shows a set of suitable colors to help them in wearing appropriate clothes.

C. Stephanidis (Ed.): HCII 2014 Posters, Part II, CCIS 435, pp. 203–208, 2014.

2 Previous Work

Different applications were developed to help visually impaired and blind people identify objects' colors in real-life. This is done by utilizing the embedded camera and screen reader on smartphones [4]. Most of these applications apply image-processing algorithms either on captured images e.g. Kolorami and Color Helper or on a video stream to achieve real-time color identification e.g. Color Identifier.

Kolorami is an iPhone mobile application developed by Comparatel especially for color blind and visually impaired people. It allows the user to take a picture or import it from gallery and then analyze its colors and show the most three colors found in the picture with its approximate percentage [5].

Another application that analyses captured pictures is Color Helper 4 Men developed by Codete for Android users. It allows the user to take a picture and asks the user to determine a specific point to be analyzed and then identify the color and give suggestions for matching colors [6]. On the other hand, GreenGar, developed Color Identifier for iPhone users, works by moving the phone camera to the object center, then the application identifies the color by displaying and pronouncing its name [7].

From the previous discussion, we can see that existing applications either require the user to capture a picture in order to determine the object color, or provide real-time color identification in local language, besides, they show some sophisticated and rarely used colors. However in our application we will give Arabic speaking users the possibility to detect all colors in real time using their own language. Moreover, unknown detected colors will be approximated to nearest popular and known colors.

3 System Overview

Our application detects colors using HSL color space where Hue (H) refers to color name, Saturation (S) refers to color fullness and (L) is the Lightness of the color [9]. The system takes a pixel from the video stream of the device camera as an input, and then analyzes it to generate the color and output it in audio and text forms. Moreover, our application permits the user to view matching colors of the detected color in order to help visually impaired people in choosing clothes to wear. Our application consists of two main functions: Color Detection and Color Matching.

3.1 Color Detection

Color Detection is the process from when the user points the device camera on an item till the color name is viewed on the screen. There are two sets of colors to detect: the first is simple colors, which was made for blind people who never saw colors before. The second is complex colors, which contains more complex colors with different scales. Color is grabbed from the camera in Red-Green-Blue (RGB) model, since RGB channels interfere with each other and its chrominance and luminance are mixed, accordingly, a small variation in lightning will affect the rates for the red, green and blue components [8][12]. Thus, “As the three components (H, S, L) are independent in HSL color space, it is more suitable for color image analysis than

RGB color space" [9]. So, we convert the pixel value from RGB to HSL before determining the color. We did this by applying the equation used for RGB to HSL conversion as Kwanchai et al. mentioned in [3].

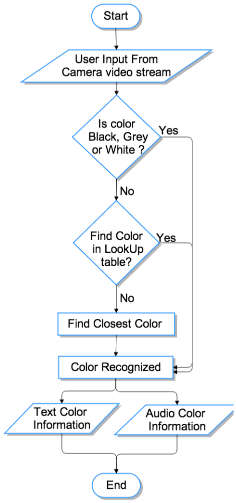

Fig. 1. Flowchart of Color Detection Process

Figure 1 illustrates the flow of our proposed system. After we have the H, S and L values for the pixel, we first check the values of saturation *S* and Luminance *L* to detect "*white*", "*black*", and "*gray*" colors. If the luminance *L* of a pixel is greater than 90, then the color is recognized as "*white*". On the contrary, if the luminance *L* of a pixel is less than 10 or (L <= 15 and S <=15) then the color is "*black*". The color is identified as "*gray*", when the saturation is less than 16 and luminance *L* matches one of these cases:

L <= 40→ Dark Grey, L <= 70→ Grey, L <= 80→ Silver and L <= 95→Light Grey

If the color is not White, Black nor gray, we search for it in the lookup table, if it is not found then: first we determine the category of the color in the pre-classified colors' categories that were classified depending on the hue value as in table 1.

Table 1. Color categories corresponding to hue values.

Hue Range	Color Category	Hue Range	Color Category
0 - 15, 346 - 360	Red	136 - 160	Spring Green
16 – 40	Orange	161 - 200	Cyan
41 – 45	Gold	201 - 255	Blue
46 – 60	Yellow	256 - 300	Purple
61 – 90	Yellow-Green	301 - 345	Pink
91 – 135	Green		

Next we calculate the difference between HSL value we obtained from the camera and the predefined HSL color values within the category. After finding the nearest value, the color name will appear on the screen and will be uttered also.

Figure 2 illustrates the color-detection interface in our system. The main window contains the video stream with point in the middle showing the exact location of the color that will be detected. Besides, there are three buttons at the bottom of the interface, button number 1 will switch between complex and simple colors, button number 2 will turn on Flash light to assist color detection in low light conditions, while button number 3 will be discussed next section.

3.2 Color Matching

Color matching function provides the user with some colors that matches a detected color in order to help the blind person in wearing suitable clothes. To accomplish this process, we used these three color schemes: complementary, split complementary and analogous to determine the matching colors [11]. Figure 3,4 illustrates an example.

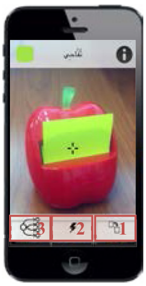

Fig. 2. Color Reader Main Interface, (1) indicates switching between complex and simple colors, (2) indicates turning on Flash light (3) indicates color matching

Fig. 3. Color Matching Interface. It shows the detected color and 4 matching colors that go with it

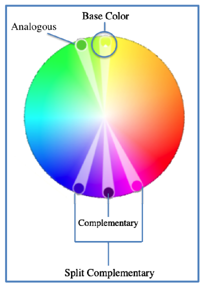

Fig. 4. Example of Color Schemes used to Match Color

4 Preliminary Evaluation

In order to demonstrate the effectiveness of our application, we discuss next the results of technical testing and user testing.

Technical testing was applied to measure the accuracy of color detection method. We tested our application under different lighting conditions (bright, natural and dark) to ensure our method can detect the correct color regardless of the light variation. We defined lighting conditions as follows: when all room lights are on, then it is bright lightening, dark lightening is defined when we depend on lampshade light only.

Table 2 illustrates the results of the technical testing. We obtained average success of 97.2% for color detection in bright lightening, 86.1% in natural lightening and 53.9% in dark lightening. However, results got worse on dark lightening where lighting conditions affects the value of the luminance (L) component of detected color.

Table 2. Color detection accuracy under different lighting conditions where 100% means the application always detects the color, 0% means the color is never detected

Color	Lighting Condition		
	Bright (%)	Natural (%)	Dark (%)
White	100	50	0
Gray	100	100	100
Black	100	100	100
Red	100	100	100
Dark Red	50	30	10
Pink	100	20	0
Fuchsia	100	100	80
Purple	100	80	70
Yellow	100	100	30
Orange	100	90	10
Brown	100	100	100
Beige	100	100	10
Green	100	100	100
Yellow-Green	100	100	70
Blue	100	100	10
Navy	100	100	30
Cyan	100	90	50
Sea Green	100	90	100
Average	97.2	86.1	53.9

Table 3. User testing results for 5 users

Tasks	Average Tasks Time (seconds)	Average Number of Errors
Identify cloth color	40	0
Turn on camera flash	36	0.4
Switch to complex color	20	0.2
Show suitable colors	33	0.2

Table 4. Average User feedback results where (1 is strongly agree and 5 is strongly disagree)

Statement	Average of user feedback results
Need for the application	1
Ease of use	1
Satisfaction	1

User Testing was applied to measure the usability of our application with end users. User testing was held at a local school for blind and visually impaired people. After introducing the application, five iPhone users (4 blind people, 1 visually impaired person) were asked to try our application and perform a set of tasks that are shown in Table 3 in order to gain an insight into how they are going to interact with the application. After that we asked them to give their feedback by rating the agreement and disagreement of some statements that is presented in Table 4 to see to what extent our application satisfies its users' needs. It is clear from the results that the application is easy to use, powerful and meets its users needs and expectations.

5 Conclusion and Future Work

In this paper we presented a real-time color identification iPhone application that helps blind and visually impaired people identify colors in real-life in an easy and effective way. Our algorithm for color detection is based on HSL color space and matching color algorithm is based on complementary, split complementary and analogous color schemes. Our evaluation showed that our application provides high detection accuracy of colors in different lighting conditions. Moreover, we found that the application is useful and satisfies its users' needs.

In order to make our application more valuable for cloth identification, we aim to improve the detection algorithm to detect cloth patterns besides colors.

References

1. Global trends in the magnitude of blindness and visual impairment, http://www.who.int/blindness/causes/trends/en/
2. Paisios, N., Subramanian, L., Rubinsteyn, A.: Choosing which Clothes to Wear Confidently: A Tool for Pattern Matching. In: Workshop on Frontiers in Accessibility for Pervasive Computing. ACM (2012)
3. Kwanchai, K., Prajin, P., Wichian, P.: Development of Object Detection Software for Mobile Robot using an AForce.Net Framework. In: Ninth International Conference on ICT and Knowledge Engineering, pp. 201–206. IEEE Thailand (2011)
4. Peng, E., Peursum, P., Li, L., Venkatesh, S.: A Smartphone-Based Obstacle Sensor for the Visually Impaired. In: Yu, Z., Liscano, R., Chen, G., Zhang, D., Zhou, X. (eds.) UIC 2010. LNCS, vol. 6406, pp. 590–604. Springer, Heidelberg (2010)
5. Kolorami, https://itunes.apple.com/us/app/kolorami/id394254215?mt=8
6. Color Helper, https://play.google.com/store/apps/details?id=co.codete.android.colorhelper
7. Color Identifier, http://itunes.apple.com/us/app/coloridentifier/id363346987?mt=8
8. Le, T.T., Tran, S.T., Mita, S., Nguyen, T.D.: Real Time Traffic Sign Detection Using Color and Shape-Based Features. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010. LNCS, vol. 5991, pp. 268–278. Springer, Heidelberg (2010)
9. Pan, R., Gao, W., Liu, J.: Color Clustering Analysis of Yarn-dyed Fabric in HSL Color Space. In: WRI World Congress on Software Engineering, WCSE 2009, vol. 2, pp. 273–278. IEEE, Xiamen (2009)
10. Tian, Y., Yuan, S.: Clothes Matching for Blind and Color Blind People. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010, Part II. LNCS, vol. 6180, pp. 324–331. Springer, Heidelberg (2010)
11. Rhyne, T.: Applying Artistic Color Theories to Visualization. In: Dill, J., et al. (eds.) Expanding the Frontiers of Visual Analytics and Visualization, pp. 263–283. Springer London (2012)
12. Toledo, F.J., Martínez, J.J., Garrigós, J., Ferrández, J.: Skin Color Detection For Real Time Mobile Applications (Spain In Field Programmable Logic and Applications). In: International Conference on FPL 2006. IEEE Spain (2006)