Color Blindness Algorithm Comparison for Developing an Android Application

By Mrs. Baswaraju Swathi[1], Koushalya R[2], Vishal Roshan J3 & Gowtham M N[4]

---***---

Abstract: Color blind is a type of Color Vision Deficiency, which is the inability that a person cannot realize the differences between some colors.They might experience difficulty in specifically spotting some of the most common colors. There are three types of color blindness: Monochromacy, Dichromacy, and Anomalous Trichromacy respectively. Color blind cannot be cured. Today, technology gets up with solutions to help individuals with color blindness to see the image and distinguish between the different yet the most basic colors using some algorithms.

This paper presents a smartphone based experimental comparison of color correction algorithms specifically designed for Dichromacy color-blind viewers: Protanopia, Duteranopia, and Tritanopia. This comparison includes LMS Daltonization algorithm, Color- blind Filter Service (CBFS) algorithm, LAB color corrector algorithm, and the shifting color algorithm respectively,to obtain results that are accurate.

The LMS algorithm is implemented for all the three types of Dichromacy. While CBFS, LAB adjustment, and Shifting color algorithms are applied to correct colors for Protanopia, Duteranopia, and Tritanopia respectively. The results show that the processing time for LMS algorithm is slow compared to other algorithms. For Protanopia condition, the LMS algorithm is better than CBFS algorithm as the LMS algorithm only changes color of confused areas with no change in the brightness.

For Duteranopia condition, the LAB color correction is better than the LMS algorithm. For Tritanopia condition, both the shifting color algorithm and the LMS algorithm may produce a new confusion in the processed images. An application interface is implemented to enable the user to choose the algorithm that gives the most appropriate results and help in the improvement of the condition.

Keywords—Color-blindness, Protanopia, Duteranopia, Tritanopia, Daltoniza- tion, LMS Detonaization, LAB color, OpenCV, Android application.

1. Introduction

Color blindness affects approximately 1 in 12 men and 1 in 200 women in the world. Most people with color blindness have an ability to see things as clearly as other people but they unable to see red, green or blue light clearly, making it difficult to make some crucial decisions in their everyday lives. There are extremely rare cases where people are unable to see any color at all there are different color blindness causes based on the colors that cannot be identified.

For most people with color vision deficiency, the condition is genetic and has been inherited from their mother, due to various genetic and hereditary factors. Also, there are some diseases causing color blindness such as diabetes and multiple sclerosis or acquiring the condition over time due to the aging process, medication and so on, making these changes from the non genetics background. In Human vision, there are two types of photoreceptors: rods and cones. Rods are sensitive to light while cones are sensitive to colors. Cones have three types; L-cones which are sensitive to long wavelength (red), M-cones which are sensitive to middle wavelength (green), and S-cones which are sensitive to short wavelength (blue).

According to these cones and the related colors, there are three types of color blindness:

1) Mono- chromacy, in which no cones or only one cone type exist in the eye.

2) Dichromacy, in which one cone type is missing, which can be classified into three types: Protanopia, in which L-cones are missing, Duteranopia, in which M-cones are missing, and Tritanopia, in which S- cones are missing.

3) Anomalous Trichromacy, is a condition in which there is a reduction in the sensitivity to a particular color, which can be of three types,they are: Protanomaly which corresponds to a reduced sensitivity to red light, Dueteranomaly which corresponds to a reduced sensitivity to green light, and Tritanomaly which corresponds to a reduced sensitivity to blue light. Protanopia and Duteranopia are the two types of red-green color blindness respectively. Tritanopia is known to be blue-yellow color blindness.

Table 1 summarizes the different color-blind types, their causes and effects.

Table 1. Color Blindness Types, Causes and Effects

Color blind type	Cause	Effect
Monochromacy	No cones or only one cone type exist.	Inability to see any color (see the world in grey shades).
Monochromacy	No cones or only one cone type exist.	Inability to see any color (see the world in grey shades).
Dichromacy	One cone type is missing, three types: 1) Protanopia: L-cones are missing. 2) Duteranopia: M-cones are missing. Tritanopia: S-cones are missing.	Inability to see the color corresponding to the missed cone type. Inability to see Red color. (Red Blind). Inability to see Green color (Green Blind). Inability to see Blue color (Blue Blind).
Anomalous Trichromacy	All cone types are exist, but they are not aligned, three types: 1) Protanomaly: L-cones are not aligned. 2) Dueteranomaly: M-cones are not aligned. 3) Tritanomaly: S-cones are not aligned.	Reduction in the sensitivity to a particular color. Less sensitivity to Red color (Red week). Less sensitivity to Green color (Green week). Less sensitivity to Blue color (Blue week).

This paper focuses on Dichromacy color blind type which is the most common type of color blindness. The vision and the abilty of the individuals affected with this condition might differ from person to person. This paper presents a smartphone based experimental comparison of color correction algorithms for Dichromacy viewers, with the results being able to help in further enhancements and advancements. This comparison includes LMS Daltonization algorithm [1], Color-blind Filter Service (CBFS) algorithm [2], LAB color corrector algorithm [3], and the shifting color algorithm [4] respectively.

The LMS algorithm is implemented for all the three types of Dichromacy which are: Protanopia, Duteranopia, and Tritanopia. The CBFS algorithm is implemented only for Protanopia condition. The LAB color corrector algorithm is implemented only for Duteranopia condition. The shifting color algorithm is implemented only for Tritanopia condition.

2. Literature Survey

In Today's advanced internet world and websites, there are many computerized aids developed for color blindness, some of the most prominent ones are:

- Color-blind Testers.
- Color-blind Simulators.
- Color correction/recognition.

2.1 Color-blind Testers

Color-blind testers are used for testing if the person has a color blind, and if yes, determine the type of color blindness.

There are many online web-based and smartphone-based applications tests that are available to test the color vision, providing us with many options and choices. Examples of the ost popular online web-based tests are: the *Ishihara Plate* *test*, which was introduced by Dr. Shinobu Ishihara from Japan, the *Farnsworth D-15 arrangement test* which was introduced by Farnsworth in 1947, the *PseudoIsochromatic Plate (PIP) Test*, and finally *Farnsworth-Munsell 100 Hue Test* which was developed by Dean Farnsworth in the 1943,all these being the most promising tests that are available in the internet. An important example of mobile applications tests in Android is *Color Blind Check*. It is important to mention that the test results of these tests may be varied with different lighting conditions and with different trials on different devices, as each computer/mobile phone screen has different color settings and this may produce varied results too.

2.2 Color-blind Simulators

Using color blindness simulators reveals how images may appear to users with a variety of color blindness conditions, making it more effective to the viewers. People use color-blind simulators to check their art work, pictures, documents and web pages for color blind visibility and to make it more accurate and precise. Some simulators are offered as an offline software program, an online web page, or a mobile application and many other options are available. Table 2 shows some examples of these simulators and their supported color blindness types and platforms that have been checked.

Table 2. Color Blind Simulator Examples

Color-blind Simulator	**The supported color blindness types**	**Platforms**
Color Oracle[3]	all types of Dichromacy: • Deuteranopia • Protanopia • Tritanopia	• a downloaded program that works offline for various operating systems like Window, Mac and Linux.
Vischeck[4]	all types of Dichromacy: • Deuteranopia • Protanopia • Tritanopia	• a website that can run on a chosen uploaded image after selecting the type of color vision to simulate and assist. • a web-page. • an offline plugin with the help of Adobe Photoshop or ImageJ for different platforms like Window, Mac and Linux.
CVD simula-tor(Coblis)5	• Monochromacy all types of Dichromacy: • Deuteranopia • Protanopia • Tritanopia all types of Anomalous Trichromacy: • Protanomaly • Deuteranomaly • Tritanomaly	• a website that can run on a chosen uploaded image after selecting the type of color vision to simulate. Support image zooming and editing features.
Color Blindness Simulate Correct6	all common colorblindness types are supported.	• an Android application that simulates and corrects color blindness in real-time using the built-in camera of the mobile phone and devices.

2.3 Color Correction for Dichromacy

There are many researched and advanced proposed color correction techniques for Dichromacy that are available. These techniques vary in terms of the supported Dicromacy types and the appropriate algorithms used. The most comonly used three algorithms: LMS Daltonization, color contrast enhancement, and LAB color adjustment. The LMS Daltonization algorithm,The mentioned color correction algorithms can be used for each type of Dichromacy color blindness. Hence, algorithms applied in the papers are suggested for a particular type of color blindness, but this is not precluded to apply them and work properly for another type,making it more efficient

3. Algorithms implementation and Experiments

3.1 LMS Daltonization Algorithm

The LMS algorithm in the developed application is implemented for Protanopia, Duteranopia, and Tritanopia conditions respectively. It is the most famous algorithm used for color-blindness correction due to reliable results. It's idea is to use the information lost in the simulation of color blindness and use LMS color space to compensate colors missing in each group/type of cones, long (L), medium (M), and short (S) in order to be predictable to the viewer and provide accurate results .

the javascript implementation of the algorithm is as follows:

```
if(!window.Color) Color = { };

if(!window.Color.Vision) Color.Vision = { };

(function() {

/*

          Color.Vision.Daltonize : v0.1

          ------------------------------

          "Analysis of Color Blindness" by Onur Fidaner, Poliang Lin and Nevran Ozguven.

          http://scien.stanford.edu/class/psych221/projects/05/ofidaner/project_report.pdf

                    "Digital Video Colourmaps for Checking the Legibility of Displays by Dichromats" by FranÃ§oise
ViÃ©not, Hans Brettel and John D. Mollon

          http://vision.psychol.cam.ac.uk/jdmollon/papers/colourmaps.pdf

*/

var CVDMatrix = { // Color Vision Deficiency

          "Protanope": [ // reds are greatly reduced (1% men)

                    0.0, 2.02344, -2.52581,

                    0.0, 1.0,     0.0,

                    0.0, 0.0,     1.0

          ],

          "Deuteranope": [ // greens are greatly reduced (1% men)

                    1.0,     0.0, 0.0,

                    0.494207, 0.0, 1.24827,

                    0.0,     0.0, 1.0

          ],

          "Tritanope": [ // blues are greatly reduced (0.003% population)

                    1.0,     0.0,     0.0,

                    0.0,     1.0,     0.0,

                    -0.395913, 0.801109, 0.0
```

```
        ]
};
Color.Vision.Daltonize = function(image, options) {
        if(!options) options = { };
        var type = typeof options.type == "string" ? options.type : "Normal",
                amount = typeof options.amount == "number" ? options.amount : 1.0,
                canvas = document.createElement("canvas"),
                ctx = canvas.getContext("2d");
        canvas.width = image.width;
        canvas.height = image.height;
        ctx.drawImage(image, 0, 0);
        try {
                var imageData = ctx.getImageData(0, 0, canvas.width, canvas.height),
                        data = imageData.data;
        } catch(e) { }
        // Apply Daltonization
        var cvdi = CVDMatrix[type],
                cvdi_a = cvd[0],
                cvdi_b = cvd[1],
                cvd_c = cvd[2],
                cvdi_d = cvd[3],
                cvdi_e = cvd[4],
                cvdi_f = cvd[5],
                cvdi_g = cvd[6],
                cvdi_h = cvd[7],
                cvdi i = cvd[8];
        var L, M, Si, l, m, s, R, G, B, RR, GG, BB;
        for(var id = 0, length = data.length; id < length; id += 4) {
                var r = data[id],
                        g = data[id + 1],
                        b = data[id + 2];
                // RGB to LMS matrix conversion
```

```
L = (17.8824 * r) + (43.5161 * g) + (4.11935 * b);

M = (3.45565 * r) + (27.1554 * g) + (3.86714 * b);

S = (0.0299566 * r) + (0.184309 * g) + (1.46709 * b);

// Simulate color blindness

l = (cvd_a * L) + (cvd_b * M) + (cvd_c * S);

m = (cvd_d * L) + (cvd_e * M) + (cvd_f * S);

s = (cvd_g * L) + (cvd_h * M) + (cvd_i * S);

// LMS to RGB matrix conversion

R = (0.0809444479 * l) + (-0.130504409 * m) + (0.116721066 * s);

G = (-0.0102485335 * l) + (0.0540193266 * m) + (-0.113614708 * s);

B = (-0.000365296938 * l) + (-0.00412161469 * m) + (0.693511405 * s);

// Isolate invisible colors to color vision deficiency (calculate error matrix)

R = r - R;

G = g - G;

B = b - B;

// Shift colors towards visible spectrum (apply error modification)

RR = (0.0 * R) + (0.0 * G) + (0.0 * B);

GG = (0.7 * R) + (1.0 * G) + (0.0 * B);

BB = (0.7 * R) + (0.0 * G) + (1.0 * B);

// Add compensation to original values

R = RR + r;

G = GG + g;

B = BB + b;

// Clamp values

if(R < 0) R = 0;

if(R > 255) R = 255;

if(G < 0) G = 0;

if(G > 255) G = 255;

if(B < 0) B = 0;

if(B > 255) B = 255;

// Record color

data[id] = R >> 0;
```

```
                data[id + 1] = G >> 0;

                data[id + 2] = B >> 0;

        }

        // Record data

        ctx.putImageData(imageData, 0, 0);

        if(typeof options.callback == "function") {

                options.callback(canvas);

        }

};

})();
```

To summarize the results and observations from the LS Daltonization algorithm are

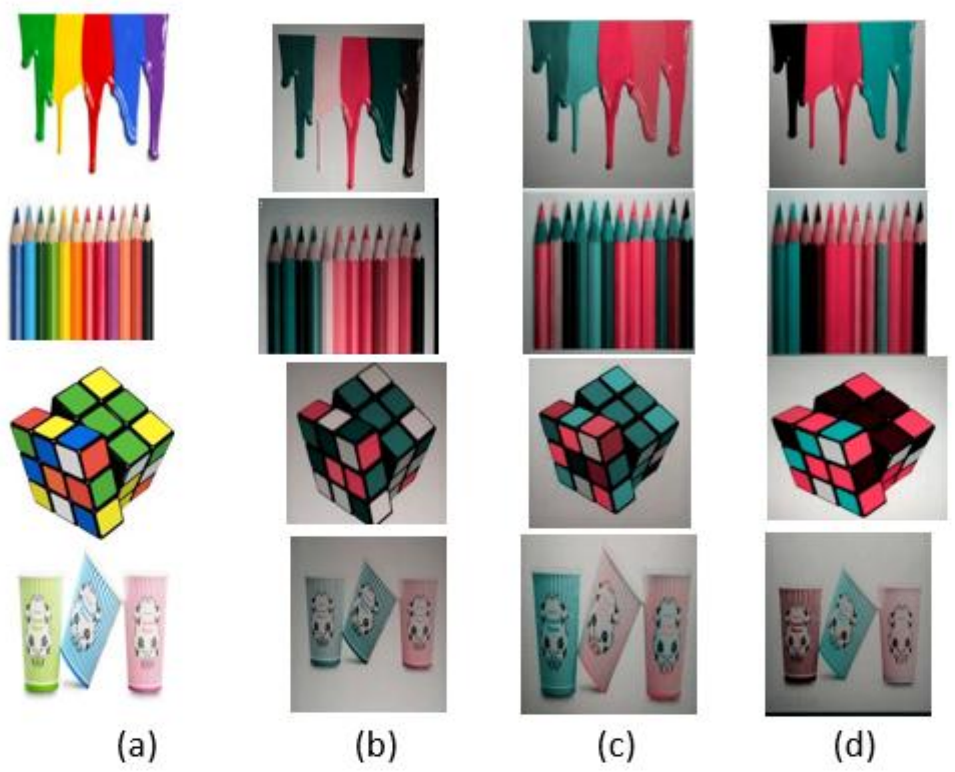


Fig. 4. LMS Daltonization algorithm Tritonopia experiments

(a) Original images with distinct color differientation.

(b) Original images as seen by Tritonopia.

(c)LMS processed images as seen by Tritonopia using the Protanopia error modification and other differences.

(d)LMS processed Images as seen by Tritonopia using the error modification.

3.2 Color-Blind Filter Service Algorithm

The following steps are the steps used to process and rectify the user provided image:

```
        //Input: RGB input image
      //Output: RGB color corrected image
 1:   Convert RGB image to HSL
          2: for each image pixel, If the pixel color is close to the dominant color of the RGB image pixels (red/green) then
                    Hue ! Hue ! 30%
                    Saturation ! Saturation ! 10%
                    Lightness ! Lightness + 25%
               else
               Saturation ! Saturation + 10%
        Lightness ! Lightness ! 10%
        3: Convert HSL image back to RGB.
```

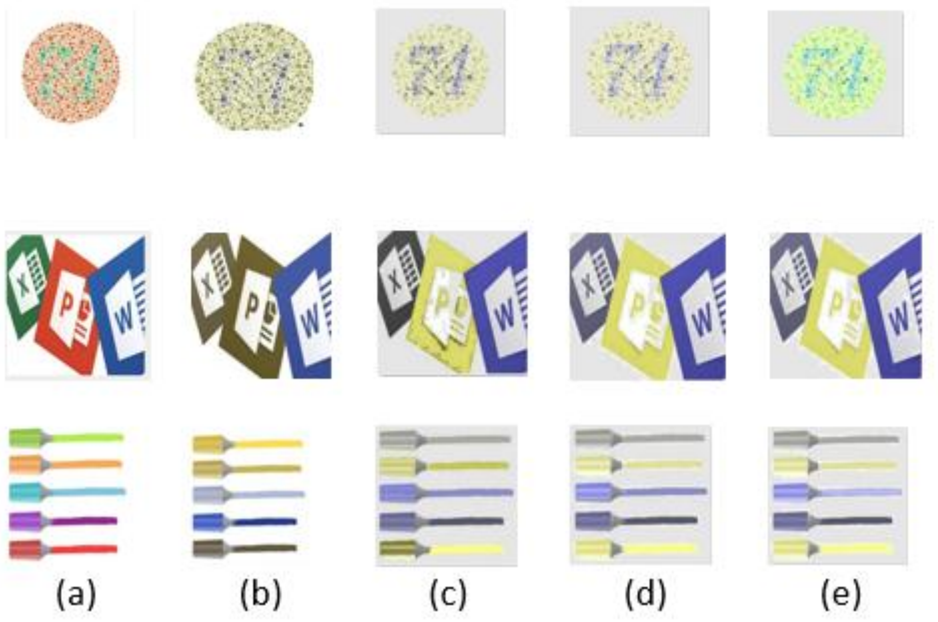


Fig. 5. The CBFS algorithm Protanopia experiments

(a)Original images.

(b) Original images as seen by Protanopia.

(c)CBFS processed Images as seen by Protanopia (Closeness parameter value =30).

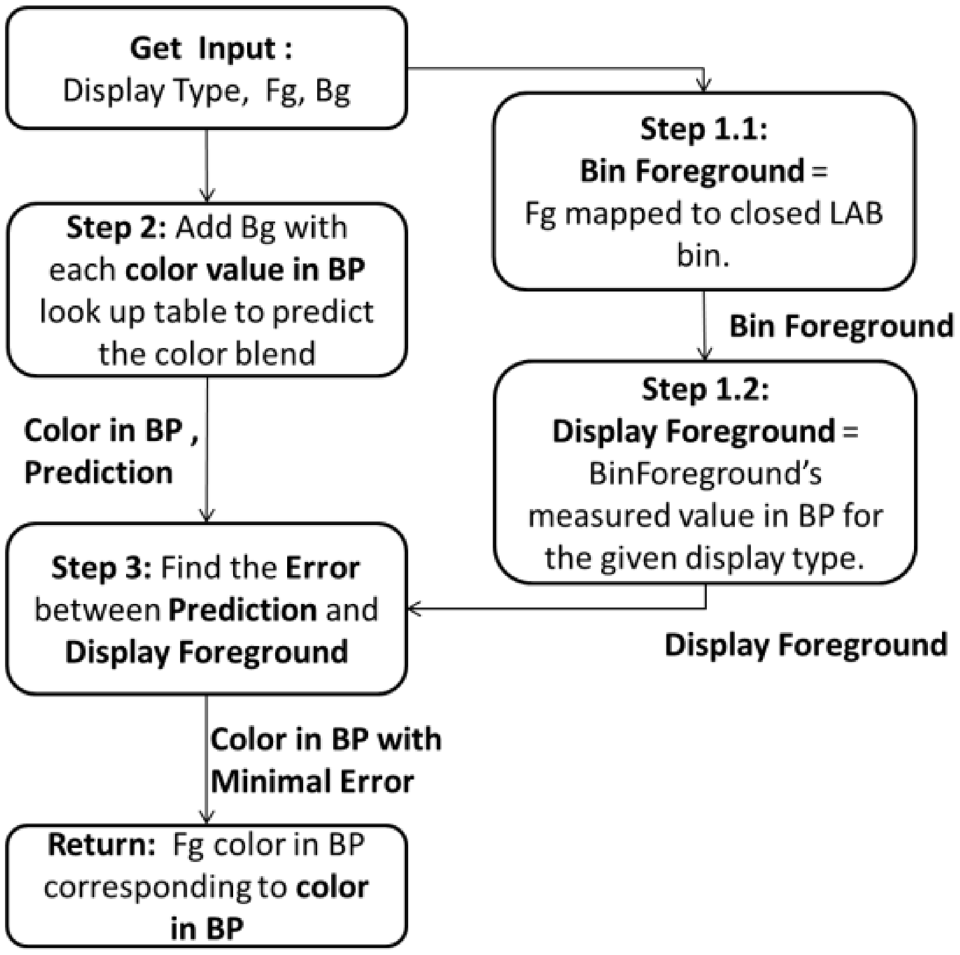
(d) CBFS processed Images as seen by Protanopia (Closeness parameter value =70).

(e)CBFS processed Images as seen by Protanopia (Closeness parameter value=90).

3.3 LAB Color Corrector

The LAB Color Corrector algorithm in the developed application is implemented for Duteranopia condition. It corrects colors for color blindness affected individuals by modifying reds and greens of an image to increase color contrast for a color blinder . The LAB Color Corrector implementation steps are listed in algorithm 3. The first operation is to adjust each pixel of A component, where a positive A means it is closer to red while negative A means it is closer to green as the LAB color range [-100, 100],depending on the pixel closeness.

However, in Android platform, LAB ranges from 0 to 255 only. Hence, we considered the half of 255 (127) as 0 the positive part, and values less than or equal 127 is the negative part. This algorithm lacks clear theoretical basis. It is based upon experimental procedures relying mostly on trial and error in the presence of a color-blind viewer and user. Hence, the best adjustments between L, B, and A components for Duteranopia is determined based on experimenting different trials. According to algorithm 3, these trials were chosen to increase the color contrast between red-green to be noticeable regards to Duteranopia condition and viewers only.


Get Input : Display Type, Fg, Bg
Step 1.1: Bin Foreground = Fg mapped to closed LAB bin.
Bin Foreground
Step 1.2: Display Foreground = BinForeground's measured value in BP for the given display type.
Display Foreground
Step 2: Add Bg with each color value in BP look up table to predict the color blend
Color in BP , Prediction
Step 3: Find the Error between Prediction and Display Foreground
Color in BP with Minimal Error
Return: Fg color in BP corresponding to color in BP

Algorithm 3: LAB Color Corrector

```
//Input: RGB input image
//Output: RGB color corrected image
```

1: Convert RGB image to LAB color space image.

2: Adjust the A values relative to its maximum, making positive values a bit more positive and negative values a bit more negative.

3: Adjust the B values relative to how green or red it is comparitively.

4: Change L pixels which is the brightness of the pixel relative to pixels A values accordingly.

5: The image is converted back to the RGB color space and concatenated to ensure that pixel values lie between zero and one in ran.

3.4 Shifting Color Algorithm

The Shifting color algorithm in the developed application is implemented for Tri- tanopia condition. In Shifting color algorithm RGB image is read and colors are shifted by different ratios in HSV color space (Hue, Saturation, and Value) . In algorithm , shifting color steps are explained. The ! value in the algorithm defines the shifting ratio and it would be changed according to the color vision deficiency type . To specify the ! value for Tritanopia algorithm, ! values are changed from 0.1 to 0.9 and tested for each one on some images as shown in Figure 6. The Hue channel in OpenCV has a range value from 0 to 179, so, the ! value is multiplied by 180. Hue values of the whole image are shifted with a fixed rate, the colors will be changed but the information will be preserved.

Algorithm 4: Shifting Color

//Input: RGB input image

//Output: RGB color corrected image

1: Convert RGB image to HSV color space image.

2: Shifting colors for each pixel (x, y) using equation

!!" ! !!" ! !

3: Check the values of hue channel to be in the range (0 to 0.9). **If** (!!">1) **then** change the hue value using Equation . **If** (!!"<0) **then** change the hue value using Equation .

!!" ! ! ! !!"

!!" ! !

4: Convert HSV image back to RGB image.

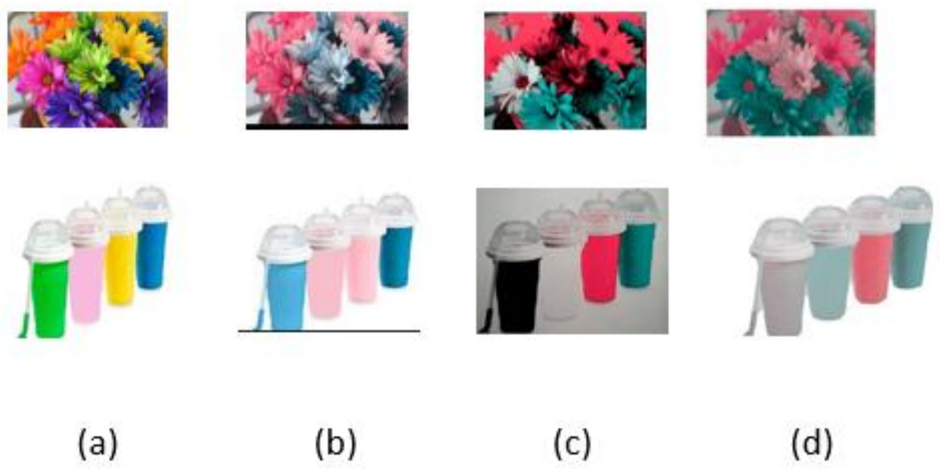
4. Conclusions

The goal of this paper is to present a smartphone based experimental comparison of color correction algorithms for Dichromacy viewers to assist them effectively. This comparison included the LMS Daltonization algorithm, Color-blind Filter Service (CBFS) algorithm, LAB color corrector algorithm, and the shifting color algorithm respectively. The LMS algorithm is implemented for all the three types of Dichromacy: Protanopia, Duteranopia, and Tritanopia. In our implementation of the LMS algorithm, a new proposed error modification method for Tritanopia condition.

The CBFS algorithm is implemented only for Protanopia conditioned viewers. The LAB color corrector algorithm is implemented only for Duteranopia conditioned viewers. The shifting color algorithm is implemented only for Tritanopia conditioned viewers. These algorithms convert colors to other colors that colorblind persons can distinguish and process effectively.

The results show that the processing time for LMS algorithm is slow compared to other algorithms. For Protanopia people, the LMS algorithm is better than CBFS algorithm as the LMS algorithm only changes color of confused areas with no change in the brightness and saturation. For Duteranopia people, the LAB color correction is better than the LMS algorithm. For Tritanopia people, both the shifting color algorithm and the LMS algorithm may produce a new confusion in the processed images.

Although these algorithms provide the end user with accurate and precise results, the algorithm to be selected must be based on the color blindness condition type and the severity of the condition.



(a) Original images

(b) Original images as seen by Tritonopia

(c) LMS processed images as seen by Tritonopia

(d) The Shifting color algorithm processed images as seen by Tritonopia.

References

[1] Deane B. Judd, "The Color Perceptions of Deuteranopic and Protanopic Observers," J.

Opt. Soc. Am. 39, 1949, pp. 252-256. https://doi.org/10.1364/JOSA.39.000252

[2] "Color Blindness," 2016. [Online]. Available: http://www.colourblindawareness.org/col our-blindness/. [Accessed 19 February 2018].

[3] B. L. Cole, "Assessment of inherited color vision defects in clinical practice," Clin Exp Optom., vol. 90(3), 2007, pp. 157-75. https://doi.org/10.1111/j.1444-0938.2007.00135.x

[4] "Types of Color Blindness," 2016. [Online]. Available: http://www.colourblindaware ness.org/colour-blindness/types-of-colour-blindness/. [Accessed 19 February 2018].

[5] J. Rumi"ski, J. Wtorek, M. Kaczmarek, A. Bujnowski, T. Kocejko and A. Poli"ski, "Image Simulation and Annotation for Color Blinded," in 2010 2nd International Conference on Information Technology, (2010 ICIT), Gdansk, 2010.

[6] Jyoti D. Badlani, C.N. Deshmukh, "A Novel Technique for Modification of Images for Deuteranopic Viewers," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 4, April 2016, pp. 467-473.

[7] N. Halder, D. Roy, T. Chowdhury, A. Chattaraj and P. Roy, "Image Color Transformation for Deuteranopia Patients using Daltonization," IOSR Journal of VLSI and Signal Pro- cessing (IOSR-JVSP), vol. 5, pp. 15-20, 2015.