Color exaggeration for dichromats using weighted edge

Daisuke Miyazaki[0000−0002−7312−0543] and Harumichi Morimoto[1]

Hiroshima City University, Hiroshima 731-3194, Japan
miyazaki@hiroshima-cu.ac.jp
http://www.info.hiroshima-cu.ac.jp/~miyazaki/

Abstract. Dichromats recognize colors using two out of three cone cells; L, M, and S. To extend the ability of dichromats to recognize the color difference, we propose a method to expand the color difference when observed by dichromats. We analyze the color between the neighboring pixels not in intensity space but chromaticity space and form a Poisson equation. In addition, we use the sigmoid function to weigh the edge of a color image. The color difference can be adequately tuned manually by the weight parameter so that the dichromats can obtain the image that they want where the visibility of the color is enhanced.

Keywords: color blindness · color vision deficiency · dichromat · recoloring · edge exaggeration

1 Introduction

Red, green, and blue colors are detected by three kinds of cone cells embedded in the retina. Dichromats use two of them to recognize colors. In this paper, we propose a method to enhance the visibility of dichromats.

Enhancing the visibility of color image for dichromats is an important research field[1, 2, 4, 6–12, 14–17, 19–26, 28–31].

Unlike these methods, our method analyzes not in color space but image space (*i.e.*, pixel coordinates) (Fig. 1). We formulate the Poisson equation so that the relative color difference between neighboring pixels will be preserved. Some methods [5, 18, 27] also solve the Poisson equation to enhance the visibility of dichromats. These methods [5, 27] form the Poisson equation in RGB intensity space, while our method forms in xy-chromaticity space. As a result, our method exaggerates the color difference between neighboring pixels. One of the existing methods [18] also forms the Poisson equation in xy-choromaticity space, however, our method can exaggerate the color difference by changing the parameter of the exaggeration weight.

Section 2 explaines the basic theory, and Section 3 shows our method. We show some results in Section 4. Section 5 concludes our paper, and we also discuss the disadvantage of our method.

[1] Current address: Hitachi Solutions West Japan, Ltd., Hiroshima, Japan

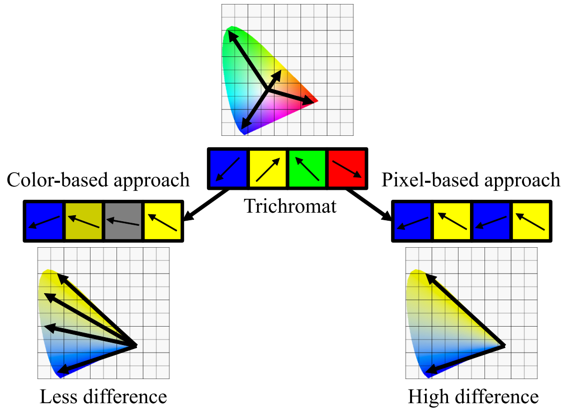

Fig. 1. Schematic explanation of color-based approach and pixel-based approach.

2 Hue for dichromats

The color value which dichromats perceive can be calculated as follows. RGB value is first converted to CIE-XYZ value, and after that, it is converted to LMS value. LMS represents the sensitivity of cone cells. The procedures to calculate the LMS values of dichromats are shown in some pieces of literature such as Judd [13] and Brettel et al. [3]. In this paper, we follow Judd [13]. The conversion formula for protanopia is shown below.

$$\begin{pmatrix} L_{\mathrm{p}} \\ M_{\mathrm{p}} \\ S_{\mathrm{p}} \end{pmatrix} = \begin{pmatrix} 0.0 & 2.02 & -2.52 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix} \begin{pmatrix} L \\ M \\ S \end{pmatrix} . \tag{1}$$

And, the conversion formula for deuteranopia is shown below.

$$\begin{pmatrix} L_{\mathrm{d}} \\ M_{\mathrm{d}} \\ S_{\mathrm{d}} \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.49 & 0.0 & 1.25 \\ 0.0 & 0.0 & 1.0 \end{pmatrix} \begin{pmatrix} L \\ M \\ S \end{pmatrix} . \tag{2}$$

In xy-diagram calcualted from CIE-XYZ value, the white color is placed in $(x, y) = (0.33, 0.33)$ for trichromats. First, the hue α of trichromats is defined as an angle defined in xy-plane (Fig. 2 (a)). The trichromatic hue α is defined as an angle around the white point $(x, y) = (0.33, 0.33)$. $0°$ of α is defined, for example, as the direction of $-45°$. The hue angle α of a certain color (x, y) is calculated as follows.

$$\alpha = \frac{\pi}{4} + \tan^{-1} \frac{y - 0.33}{x - 0.33} . \tag{3}$$

The hue β of dichromats (Fig. 2 (b)–(c)) has strong relation with the L*a*b* hue of trichromats [19]. Following Judd [13], the white point of protanopia is

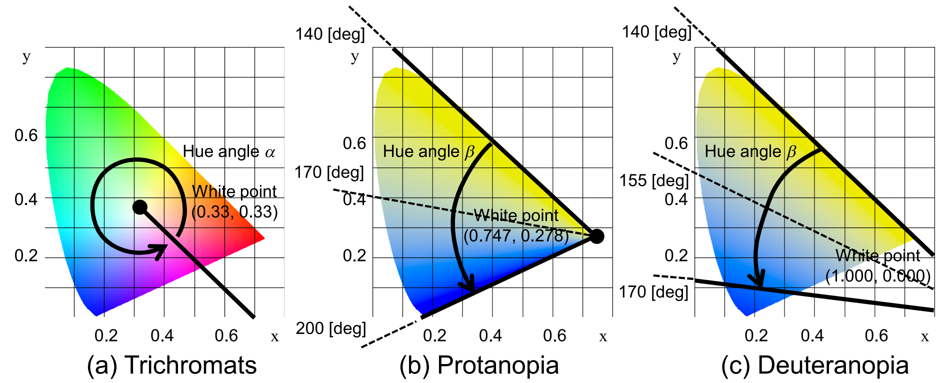

Fig. 2. Definition of hue of (a) trichromats, (b) protanopia, and (c) deuteranopia.

$(x, y) = (0.747, 0.253)$ and that of deuteranopia is $(x, y) = (1.000, 0.000)$. The hue β is defined as the angle around these white points, where it ranges from 140° direction to 200° direction for protanopia and ranges from 140° direction to 170° direction for deuteranopia. The hue angle β of protanopia is calculated as follows.

$$\beta = \frac{\pi}{180}\left(140 + \frac{\alpha}{2\pi}(200 - 140)\right) , \tag{4}$$

and the hue angle β of deuteranopia is calcuted as follows.

$$\beta = \frac{\pi}{180}\left(140 + \frac{\alpha}{2\pi}(170 - 140)\right) . \tag{5}$$

Here, the hue angle α ranges from 0 to 360°. The direction of 0° in hue angle α is casually defined (*i.e.*, 45°); however, it does not matter for our purpose as explained mathematically in Sec. 3. Sec. 3 explains that our method uses the relative value for representing the hue instead of the absolute value. This is because that the hue of trichromats ranges from 0 to 360°, but the 360° is cyclically connected to 0, and because that the hue of dichromats ranges in a limited range.

3 Color enhancement for dichromat

The purpose of the method is to enhance the visibility of the image for dichromats. As shown in Section 2, we represent the color as the hue angle shown in Fig. 2.

sRGB value of the input image is converted to CIE-XYZ. After that, CIE-XYZ value is converted to xy-chromaticity as follows.

$$\tilde{x} = \frac{\tilde{X}}{\tilde{X} + \tilde{Y} + \tilde{Z}} , \tag{6}$$

$$\tilde{y} = \frac{\tilde{Y}}{\tilde{X} + \tilde{Y} + \tilde{Z}} , \tag{7}$$

$$\tilde{z} = \frac{\tilde{Z}}{\tilde{X} + \tilde{Y} + \tilde{Z}} . \tag{8}$$

The vector from the white point $(1/3, 1/3)$ of xy chromaticity to the chromaticity of image pixel is represented as Eq. (9).

$$\mathbf{a}(u, v) = \begin{pmatrix} \tilde{x}(u, v) - 0.33 \\ \tilde{y}(u, v) - 0.33 \\ 0 \end{pmatrix} . \tag{9}$$

Here, we use (u, v) for representing the x and y components of pixel position represented in Euclidean coordinates with x and y axes.

We deonte the 4-neighbor pixel position as $(u+\Delta v, u+\Delta v)$, where the integer values Δu and Δv obey $|\Delta u| + |\Delta v| = 1$. The color vectors of neighboring pixels are also calculated as Eq. (10).

$$\tilde{\mathbf{a}}(u + \Delta u, v + \Delta v) = \begin{pmatrix} \tilde{x}(u + \Delta u, v + \Delta v) - 0.33 \\ \tilde{y}(u + \Delta u, v + \Delta v) - 0.33 \\ 0 \end{pmatrix} . \tag{10}$$

We normalize these vectors and denote them as $\hat{\mathbf{a}}(u, v)$ and $\hat{\mathbf{a}}(u+\Delta u, v+\Delta v)$ (Fig. 3). We denote the cross product of these two vectors as $\mathbf{a}$.

$$\mathbf{a}(u + \Delta u, v + \Delta v) = \hat{\mathbf{a}}(u + \Delta u, v + \Delta v) \times \hat{\mathbf{a}}(u, v) . \tag{11}$$

Calculating the arcsine of $\mathbf{a}$ results in the signed angle between $\hat{\mathbf{a}}(u+\Delta u, v+\Delta v)$ and $\hat{\mathbf{a}}(u, v)$. We denote this angle as $\Delta\alpha(u + \Delta u, v + \Delta v)$ (Fig. 4).

$$\Delta\alpha(u + \Delta u, v + \Delta v) = \sin^{-1}(\mathbf{a}(u + \Delta u, v + \Delta v)) . \tag{12}$$

The discretized representation of the Laplacian of the hue angle $\tilde{\beta}$ for dichromats is as follows.

$$\begin{aligned} \triangle\tilde{\beta}(u, v) = -\Big\{ & \tilde{\beta}(u, v) \\ & -\frac{1}{4}\tilde{\beta}(u - 1, v) - \frac{1}{4}\tilde{\beta}(u + 1, v) - \frac{1}{4}\tilde{\beta}(u, v - 1) - \frac{1}{4}\tilde{\beta}(u, v + 1)\Big\} . \end{aligned} \tag{13}$$

Same goes to α.

$$\begin{aligned} \triangle\alpha(u, v) = -\Big\{ & \alpha(u, v) \\ & -\frac{1}{4}\alpha(u - 1, v) - \frac{1}{4}\alpha(u + 1, v) - \frac{1}{4}\alpha(u, v - 1) - \frac{1}{4}\alpha(u, v + 1)\Big\} . \end{aligned} \tag{14}$$

Eq. (14) is also represented as follows.

$$\begin{aligned} \triangle\alpha(u, v) = -\Big\{ & \frac{1}{4}\left(\alpha(u, v) - \alpha(u - 1, v)\right) + \frac{1}{4}\left(\alpha(u, v) - \alpha(u + 1, v)\right) \\ & + \frac{1}{4}\left(\alpha(u, v) - \alpha(u, v - 1)\right) + \frac{1}{4}\left(\alpha(u, v) - \alpha(u, v + 1)\right)\Big\} . \end{aligned} \tag{15}$$

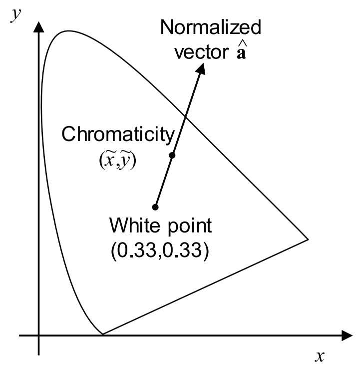

Fig. 3. Chromaticitiy vector.

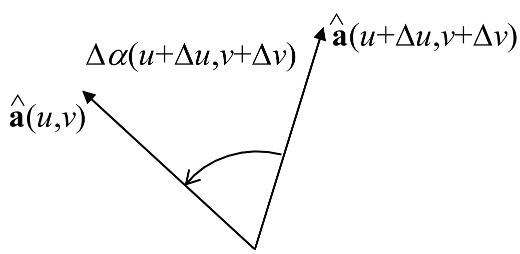

Fig. 4. Relative hue angle between two neighboring pixels.

If we simply subtract two angles, the calculation will fail since the angle has a cycle of 360°. For example, 5° minus 355° should be 10°, not −350°. If we convert an angle to a vector and calculate the angle between two vectors, this problem will not occur. Unlike dot product of two vectors, cross product of two vectors can calculate the angle with signed value. Using Eq. (11), Eq. (15) can be rewritten as follows.

$$\begin{aligned}&\triangle\alpha(u,v) = \\ &-\left\{\frac{1}{4}\Delta\tilde{\alpha}(u-1,v)+\frac{1}{4}\Delta\tilde{\alpha}(u+1,v)+\frac{1}{4}\Delta\tilde{\alpha}(u,v-1)+\frac{1}{4}\Delta\tilde{\alpha}(u,v+1)\right\}\end{aligned} \tag{16}$$

The difference of hue angle $\tilde{\beta}$ between neighboring pixels should be proportional to the difference of hue angle α between neighboring pixels. Namely, the Laplacian of $\tilde{\beta}$ should be the same as the Laplacian of α (Fig. 5), scaled with a certain constant value.

$$\triangle\tilde{\beta}(u,v) = \triangle\alpha(u,v)\,. \tag{17}$$

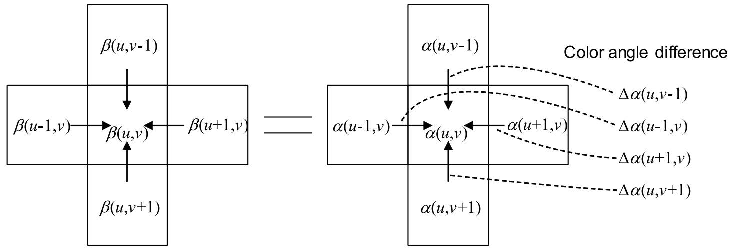

Fig. 5. Color difference between neighboring pixels.

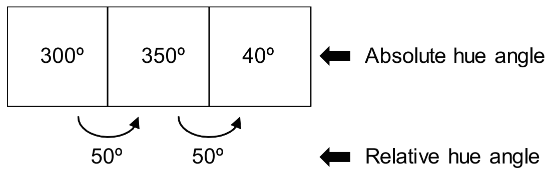

Fig. 6. Specific example of absolute and relative hue angles.

This type of formula is called Poisson equation. From Eq. (13) and Eq. (16), Eq. (17) is represented as follows.

$$\begin{aligned}&\tilde{\beta}(u,v) - \frac{1}{4}\tilde{\beta}(u-1,v) - \frac{1}{4}\tilde{\beta}(u+1,v)\\&-\frac{1}{4}\tilde{\beta}(u,v-1) - \frac{1}{4}\tilde{\beta}(u,v+1) =\\&\frac{1}{4}\Delta\tilde{\alpha}(u-1,v) + \frac{1}{4}\Delta\tilde{\alpha}(u+1,v) + \frac{1}{4}\Delta\tilde{\alpha}(u,v-1) + \frac{1}{4}\Delta\tilde{\alpha}(u,v+1)\,. \quad (18)\end{aligned}$$

The angle $\tilde{\beta}$ between neighboring pixels will become the same as the angle α between neighboring pixels if we solve Eq. (18). Although $\tilde{\beta}$ becomes similar to α, the calculated $\tilde{\beta}$ becomes free from the cycle of 360°. Unlike an identity equation $\tilde{\beta} = \alpha$ which copies the absolute angle, Eq. (17) preserves the relative angle among neighboring pixels.

Suppose that the image is consisted of three pixels, and has hue angles α which are 300°, 350°, and 40° (Fig. 6). The colors for trichromats will be blue, purple, and red. If we simply map these angles to the angle $\tilde{\beta}$, the color for dichromats becomes faint blue, deep blue, and yellow. However, if we solve the abovementioned Poisson equation, the calculated angle will be 300°, 350°, and 400°. If we map these angles to the angle $\tilde{\beta}$, for example, to 150°, 160°, and 170°, the color for dichromats becomes faintly yellow color, faintly cyan color, and faintly blue color. The color difference between neighboring pixels will be preserved if we solve the Poisson equation.

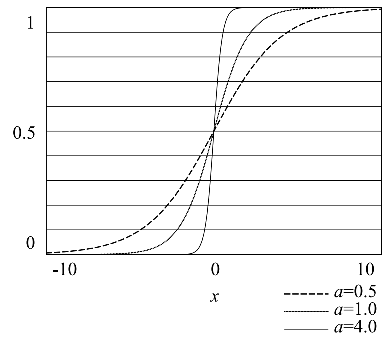

Fig. 7. Sigmoid function.

Unlike existing method [18], our method weights the color difference $\Delta\tilde{\alpha}$ of 4-neighbor using the sigmoid function (Fig. 7) subtracted by 0.5 (Eq. (19)).

$$f(x) = \frac{1}{1+\exp(-ax)} - \frac{1}{2}\,. \tag{19}$$

The real number a is called gain. Sigmoid function $(0 < y < 1)$ (Fig. 7) is a monotonically increasing function which is symmetric at the point $(0, 0.5)$. The sigmoid function becomes sharp if the gain is large, and becomes smooth if that is small. Therefore, the gain can adjust the color exaggeration.

The sigmoid function $f(\Delta\tilde{\alpha}(u+\Delta u, v+\Delta v))$ when the gain is a and the color difference is $\Delta\tilde{\alpha}(u+\Delta u, v+\Delta v)$ is shown in Eq. (20).

$$f(\Delta\tilde{\alpha}(u+\Delta u, v+\Delta v)) = \frac{1}{1+\exp(-a\Delta\tilde{\alpha}(u+\Delta u, v+\Delta v))} - \frac{1}{2}\,. \tag{20}$$

The Poisson equation (Eq. (18)) considering the sigmoid function becomes Eq. (21).

$$\begin{aligned}\tilde{\beta}(x,y) - \tfrac{1}{4}\tilde{\beta}(u,v-1) - \tfrac{1}{4}\tilde{\beta}(u-1,v) - \tfrac{1}{4}\tilde{\beta}(u+1,v) - \tfrac{1}{4}\tilde{\beta}(u,v+1)\\ = \tfrac{1}{4}f(\Delta\tilde{\alpha}(u,v-1)) + \tfrac{1}{4}f(\Delta\tilde{\alpha}(u-1,v))\\ + \tfrac{1}{4}f(\Delta\tilde{\alpha}(u+1,v)) + \tfrac{1}{4}f(\Delta\tilde{\alpha}(u,v+1))\,.\end{aligned} \tag{21}$$

The closed-form solution to $\tilde{\beta}$ (Eq. (21)) can be obtained using the LU decomposition implemented in sparse matrix library.

The followings explain the process to reconvert the angle $\tilde{\beta}$ to an RGB image.

We create the cumulative histogram of $\tilde{\beta}$. Using this cumulative histogram, we converted the angle $\tilde{\beta}$ to the angle $\hat{\beta}$ (Fig. 8). The distance from the white point of protanopia is denoted as p and that from the white point of deuteranopia is denoted as d in Eqs. (22)–(23).

$$\begin{aligned}x_p(u,v) &= p\cos\hat{\beta} + 0.747\,,\\ y_p(u,v) &= p\sin\hat{\beta} + 0.278\,,\\ z_p(u,v) &= 1.0 - x_p(u,v) - y_p(u,v)\,.\end{aligned} \tag{22}$$

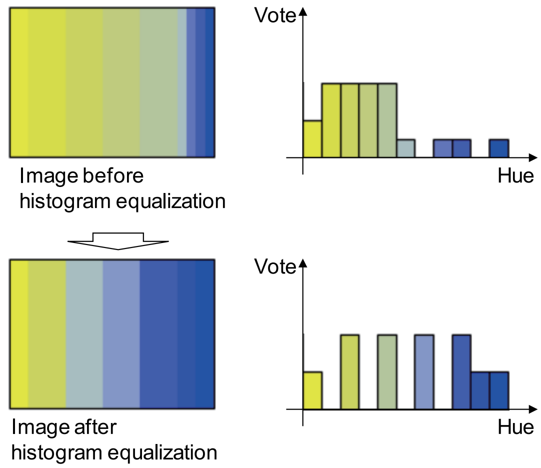

Fig. 8. Histogram equalization of dichromatic hue.

$$
\begin{aligned}
x_d(u,v) &= d\cos\hat{\beta} + 1.000\,,\\
y_d(u,v) &= d\sin\hat{\beta} + 0.000\,,\\
z_d(u,v) &= 1.0 - x_d(u,v) - y_d(u,v)\,.
\end{aligned} \tag{23}
$$

We set p and d as follows. We denote the original xy-chromaticities of the interest pixel as $\overline{x}(u,v)$ and $\overline{y}(u,v)$, and the post-processed xy-chromaticities as $x(u,v)$ and $y(u,v)$. The following t minimizes the difference between the input chromaticity and the output chromaticity.

$$
\begin{pmatrix} \overline{x}(u,v) \\ \overline{y}(u,v) \end{pmatrix} = \begin{pmatrix} C_{\mathrm{x}} \\ C_{\mathrm{y}} \end{pmatrix} + \begin{pmatrix} \cos\hat{\beta}(u,v) \\ \sin\hat{\beta}(u,v) \end{pmatrix} t\,. \tag{24}
$$

Therefore, the following is derived by solving Eq. (24).

$$
t = (\overline{x} - C_{\mathrm{x}})\cos\hat{\beta} + (\overline{y} - C_{\mathrm{y}})\sin\hat{\beta}\,. \tag{25}
$$

Note that, the white point of protanopia is defined as $(C_{\mathrm{x}}, C_{\mathrm{y}}) = (0.747, 0.278)$, and the white point of deuteranopia is defined as $(C_{\mathrm{x}}, C_{\mathrm{y}}) = (1.000, 0.000)$. From estimated t, Eq. (22), and Eq. (23), the final values of x, y, z are determined. This process is schematically depicted in Fig. 9.

At the beginning of our method, we convert the RGB of input image to XYZ, and calculate $W(u,v) = X(u,v) + Y(u,v) + Z(u,v)$. Eq. (26) converts the chromaticities $x(u,v)$, $y(u,v)$, and $z(u,v)$ calculated from the hue β to the XYZ values $\overline{X}$, $\overline{Y}$, and $\overline{Z}$.

$$
\begin{aligned}
\overline{X}(u,v) &= x(u,v)W(u,v)\,,\\
\overline{Y}(u,v) &= y(u,v)W(u,v)\,,\\
\overline{Z}(u,v) &= z(u,v)W(u,v)\,.
\end{aligned} \tag{26}
$$

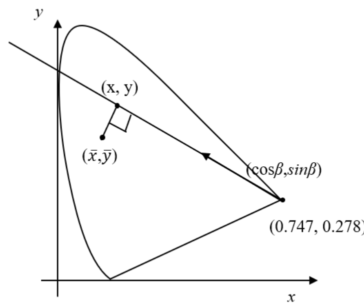

Fig. 9. Post-processing for protanopia.

Input image Protanopia's apperance Our result

Fig. 10. Our result (protanopia) [symbolic illustration].

XYZ values are obtained now, and we convert it to the dichromats' view and convert its XYZ to RGB.

4 Experiment

Fig. 10 shows one example of the result. The result of our method can represent the symbolic structure rather than the appearance of protanopia.

The dichromat's appearance of Fig. 11 (a) is shown in Fig. 11 (b)–(c). Note that some of the patches are indistinguishable. The output image of the gain $a = 0.1$ is shown in Fig. 11 (d)–(e), that of the gain $a = 5$ is shown in Fig. 11 (f)–(g), and that of the gain $a = 100$ is shown in Fig. 11 (h)–(i). Fig. 11 (d)–(i) show that the gain can adjust the amount of exaggeration of the color difference thanks to the characteristics of the sigmoid function.

The dichromat's appearance of Fig. 12 (a) is shown in Fig. 12 (b)–(c). The output image without gain is shown in Fig. 12 (d)–(e), that of the gain $a = 0.1$ is shown in Fig. 12 (f)–(g), that of the gain $a = 5$ is shown in Fig. 12 (h)–(i), and

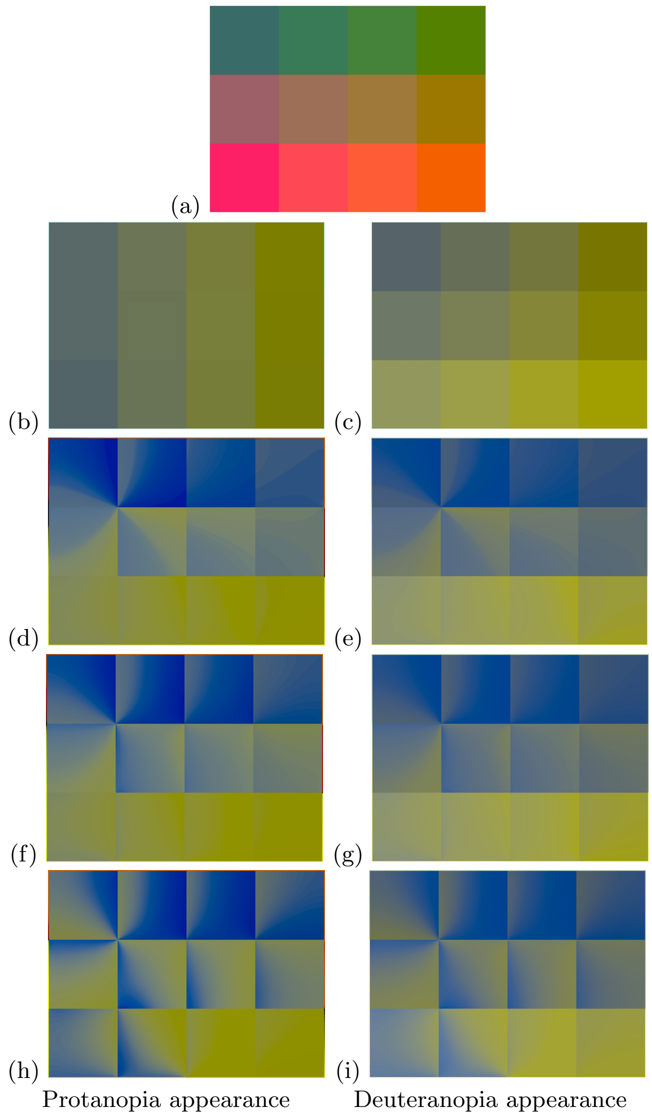

Fig. 11. Result [color patch]: (a) input, (b) perceived color of protanopia, (c) perceived color of deuteranopia, (d) our result of gain 0.1 (protanopia), (e) our result of gain 0.1 (deuteranopia), (f) our result of gain 5 (protanopia), (g) our result of gain 5 (deuteranopia), (h) our result of gain 100 (protanopia), and (i) our result of gain 100 (deuteranopia).

that of the gain $a = 100$ is shown in Fig. 12 (j)–(k). Our method can distinguish red and green leaves.

The dichromat's appearance of Fig. 13 (a) is shown in Fig. 13 (b)–(c). The output image without gain is shown in Fig. 13 (d)–(e), that of the gain $a = 0.1$ is shown in Fig. 13 (f)–(g), that of the gain $a = 5$ is shown in Fig. 13 (h)–(i), and that of the gain $a = 100$ is shown in Fig. 13 (j)–(k). If we set the gain small, the original appearance is preserved, while if we set the gain large, the color differences become large. This result empirically proves the benefit of our method compared to the methods without gain (such as Miyazaki's method [18]). Miyazaki's method does not weigh by sigmoid function, while we can exaggerate the color difference by sigmoid function.

Other results are shown in Fig. 14.

The adequate amount of color exaggeration depends on the user, the situation, and the purpose. Strong exaggeration is favored in some cases, while slight exaggeration (with naturalness preserved) is favored in other cases. As a result, the proposed method is beneficial for actual purposes, which can tune the amount of color exaggeration.

As shown in our experiments, our method successfully works for both artificial images and natural images. Our method is not sensitive to the image types, whether it is an artificial image or a natural image. However, our method is sensitive to achromatic pixels. Our method exaggerates the color difference between neighboring pixels. Therefore, if the neighboring pixel is black, gray, or white, we cannot represent the color difference between neighboring pixels, and thus, the result images might not have satisfactory color representation.

5 Conclusion

In this paper, we have proposed a method that enhances the visibility of dichromats. Our method converts the color of an image so that the image will be clear for dichromats. We have formulated the color difference of trichromat as a Poisson equation and solved it to preserve the color difference which can also be perceived by dichromats. The Poisson equation formulated in chromaticity space exaggerates the color difference of neighboring pixels, and at the same time, it preserves the chromaticity difference of trichromats. Experimental results show that our method is robust and beneficial. Also, our result has shown that the exaggeration of the color difference for dichromats can be tuned by the weighting parameter. The disadvantage of our method is that it cannot be applied to the achromatic images (black, gray, and white). Some kind of preprocessing or postprocessing may avoid such a problem, however, such processes do not fundamentally solve the problem. We are planning to theoretically solve this problem in the future.

References

1. S. Bao, G. Tanaka, H. Tamukoh, and N. Suetake, "Improvement of lightness modification method based on Craik-O'Brien effect for dichromats," in *Int'l Symp.*

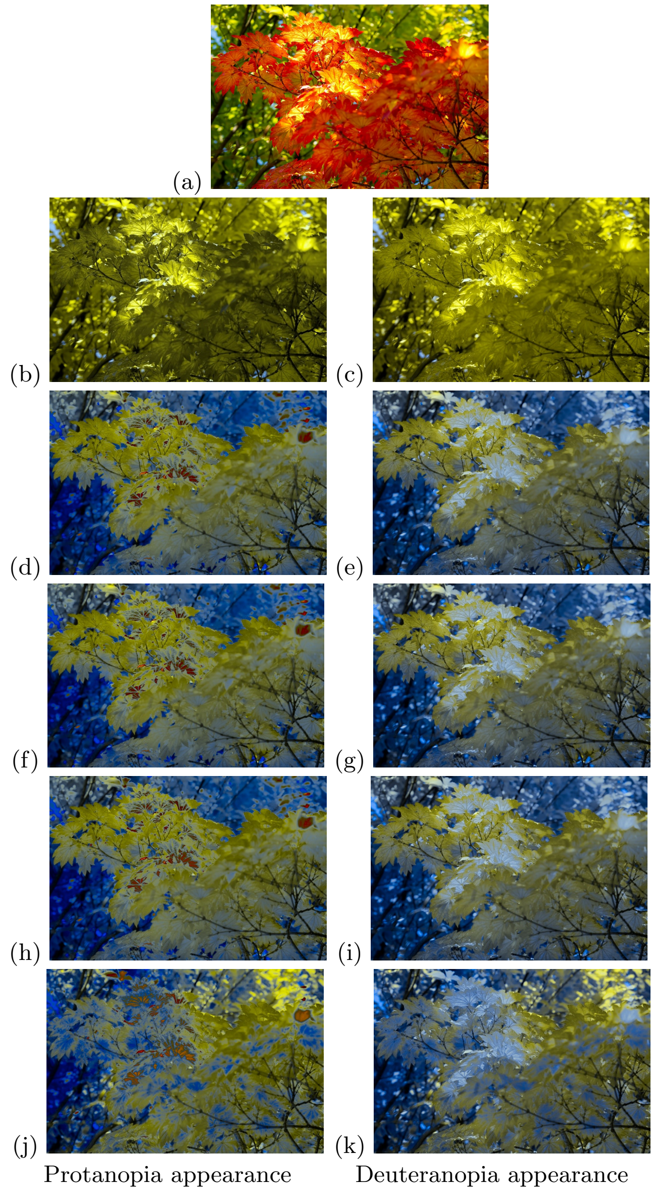

Fig. 12. Result [leaf]: (a) input, (b) perceived color of protanopia, (c) perceived color of deuteranopia, (d) previous result without gain (protanopia), (e) previous result without gain (deuteranopia), (f) our result of gain 0.1 (protanopia), (g) our result of gain 0.1 (deuteranopia), (h) our result of gain 5 (protanopia), (i) our result of gain 5 (deuteranopia), (j) our result of gain 100 (protanopia), and (k) our result of gain 100 (deuteranopia).

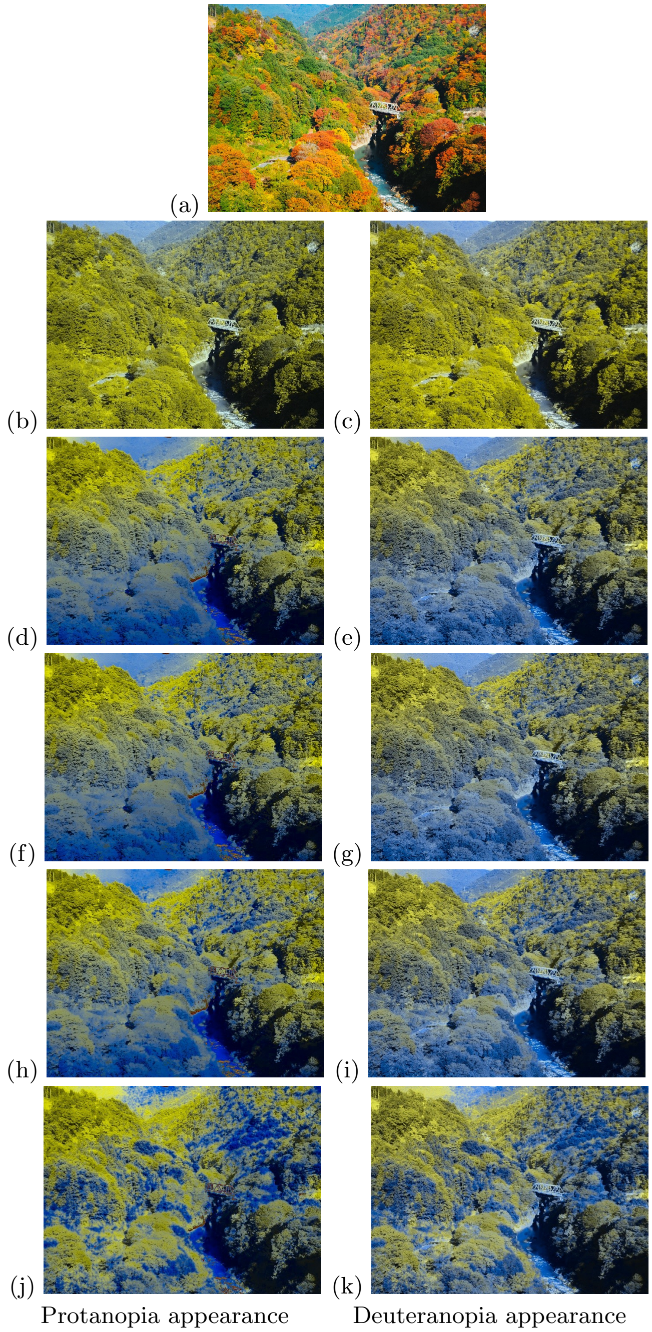

Fig. 13. Result [mountain]: (a) input, (b) perceived color of protanopia, (c) perceived color of deuteranopia, (d) previous result without gain (protanopia), (e) previous result without gain (deuteranopia), (f) our result of gain 0.1 (protanopia), (g) our result of gain 0.1 (deuteranopia), (h) our result of gain 5 (protanopia), (i) our result of gain 5 (deuteranopia), (j) our result of gain 100 (protanopia), and (k) our result of gain 100 (deuteranopia).

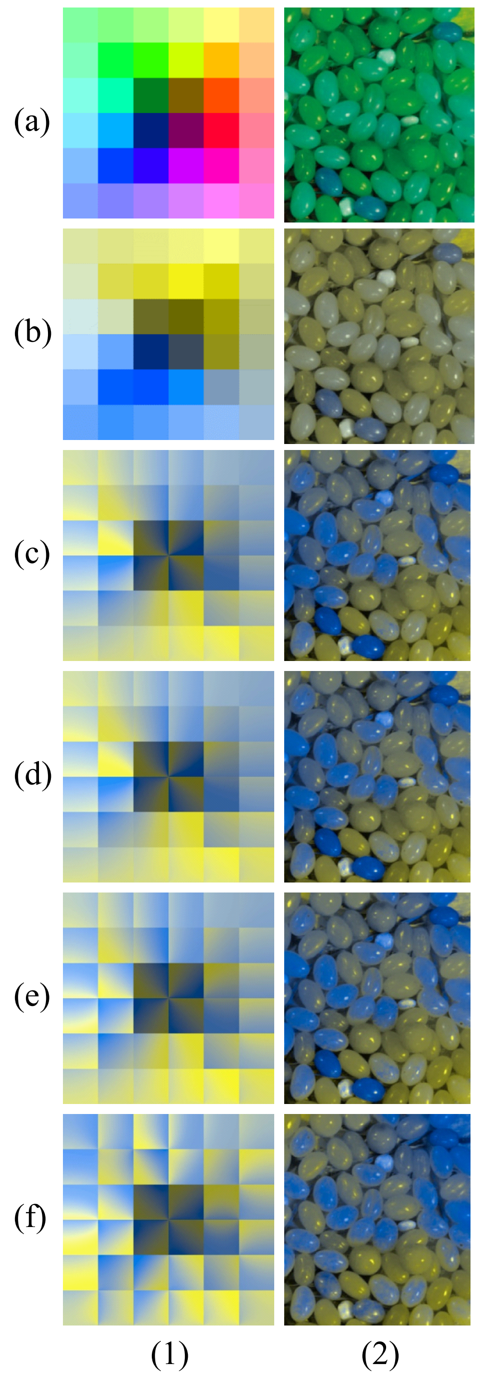

Fig. 14. Result [(1) patch, (2) beans]: (a) input, (b) perceived color of deuteranopia, (c) previous result without gain (deuteranopia), (d) our result of gain 0.1 (deuteranopia), (e) our result of gain 5 (deuteranopia), (f) our result of gain 100 (deuteranopia).

Intelligent Signal Processing and Communication Systems, pp. 75–78, 2015.
2. J. Bao, Y. Wang, Y. Ma, and X. Gu, "Re-coloring images for dichromats based on an improved adaptive mapping algorithm," in *Int'l Conf. Audio, Language and Image Processing*, pp. 152–156, 2008.
3. H. Brettel, F. Viénot, and J. D. Mollon, "Computerized simulation of color appearance for dichromats," *J. Opt. Soc. Am. A*, vol. 14, no. 10, pp. 2647–2655, 1997.
4. W. Chen, W. Chen, and H. Bao, "An efficient direct volume rendering approach for dichromats," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2144–2152, 2011.
5. I. Farup, "Individualised halo-free gradient-domain color image daltonisation," *Journal of Imaging*, vol. 6, no. 116, 10 pages, 2020.
6. M. Flinkman and S. Nakauchi, "Illuminations that improve color discrimination ability of people with red-green color vision deficiency," *J. Optical Society of America A*, vol. 34, no. 10, pp. 1914–1923, 2017.
7. C. Huang, K. Chiu, and C. Chen, "Temporal color consistency-based video reproduction for dichromats," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 950–960, 2011.
8. J. Huang, C. Chen, T. Jen, and S. Wang, "Image recolorization for the colorblind," in *IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, pp. 1161–1164, 2009.
9. J. Huang, Y. Tseng, S. Wu, and S. Wang, "Information preserving color transformation for protanopia and deuteranopia," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 711–714, 2007.
10. J.-B. Huang, S.-Y. Wu, and C.-S. Chen, "Enhancing color representation for the color vision impaired," in *Workshop on Computer Vision Applications for the Visually Impaired*, 2008.
11. M. Ichikawa, K. Tanaka, S. Kondo, K. Hiroshima, K. Ichikawa, S. Tanabe, and K. Fukami, "Preliminary study on color modification for still images to realize barrier-free color vision," in *Int'l Conf. Systems, Man and Cybernetics*, pp. 36–41, 2004.
12. J. Jeong, H. Kim, Y. Kim, T. Wang, and S. Ko, "Enhanced re-coloring method with an information preserving property for color-blind person," in *Int'l Conf. Consumer Electronics*, pp. 600–601, 2012.
13. D. B. Judd, "Standard response functions for protanopic and deuteranopic vision," *J. Optical Society of America*, vol. 35, pp. 199–221, 1945.
14. G. R. Kuhn, M. M. Oliveira, and L. A. F. Fernandes, "An efficient naturalness-preserving image-recoloring method for dichromats," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1747–1754, 2008.
15. Y. Ma, E. Wang, and Y. Wang, "An embedded image processing device for color vision deficiency," in *Int'l Conf. Biomedical Engineering and Informatics*, pp. 1041–1045, 2011.
16. M. Meng and G. Tanaka, "Proposal of minimization problem based lightness modification for protanopia and deuteranopia," in *Int'l Symp. Intelligent Signal Processing and Communication Systems*, pp. 1–6, 2016.
17. G. W. Meyer and D. P. Greenberg, "Color-defective vision and computer graphics displays," *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 28–40, 1988.
18. D. Miyazaki and S. Fujimura, "Hue enhancement for dichromats using Poisson equation," *Journal of Imaging Science and Technology*, vol. 66, no. 1, article no. 010502, pp. 1–8, 2022.

19. D. Miyazaki, S. Taomoto, and S. Hiura, "Extending the visibility of dichromats using histogram equalization of hue value defined for dichromats," *International Journal of Image and Graphics*, vol. 19, no. 3, article no. 1950016, 13 pages, 2019.
20. S. Nakauchi and T. Onouchi, "Detection and modification of confusing color combinations for red-green dichromats to achieve a color universal design," *Color Research & Application*, vol. 33, no. 3, pp. 203–211, 2008.
21. M. M. Oliveira, "Towards more accessible visualizations for color-vision-deficient indificuals," *Computing in Science & Engineering*, vol. 15, no. 5, pp. 80–87, 2013.
22. K. Rasche, R. Geist, and J. Westall, "Detail preserving reproduction of color images for monochromats and dichromats," *IEEE Computer Graphics and Applications*, vol. 25, no. 3, pp. 22–30, 2005.
23. M. G. Ribeiro and A. J. P. Gomes, "A skillet-based recoloring algorithm for dichromats," in *Int'l Conf. e-Health Networking, Applications & Services*, pp. 702–706, 2013.
24. M. Ribeiro and A. J. P. Gomes, "Recoloring algorithms for colorblind people: a survey," *ACM Computing Surveys*, vol. 52, no. 4, 37 pages, 2019.
25. J. Ruminski, J. Wtorek, J. Ruminska, M. Kaczmarek, A. Bujnowski, T. Kocejko, and A. Polinski, "Color transformation methods for dichromats," in *3rd Conf. Human System Interactions*, pp. 634–641, 2010.
26. B. Sajadi, A. Majumder, M. M. Oliveira, R. G. Schneider, and R. Raskar, "Using patterns to encode color information for dichromats," *IEEE Trans. Visualization and Computer Graphics*, vol. 19, no. 1, pp. 118–129, 2013.
27. J. T. Simon-Liedtke and I. Farup, "Multiscale daltonization in the gradient domain," *Journal of Perceptual Imaging*, vol. 1, no. 1, pp. 010503-1–010503-12, 2018.
28. J. P. Srividhya, P. Sivakumar, and M. Rajaram, "The color blindness removal technique in image by using gradient map method," in *Int'l Conf. Signal Processingg, Communication, Computing and Networking Technologies*, pp. 24–29, 2011.
29. G. Tanaka, N. Suetake, and E. Uchino, "Lightness modification of color image for protanopia and deuteranopia," *Optical Review*, vol. 17, no. 1, pp. 14–23, 2010.
30. G. Tennenholtz and I. Zachevsky, "Natural contrast enhancement for dichromats using similarity maps," in *Int'l Conf. Science of Electrical Engineering*, pp. 1–5, 2016.
31. F. Yan, G. Liu, and Y. Zheng, "An HTML5 based discoloration algorithm for people with low vision," in *Int'l Conf. Biomedical Engineering and Informatics*, pp. 837–842, 2015.