Image Enhancement for Dichromats by Weighting the Color Difference Between Neighboring Pixels

Daisuke Miyazaki
Faculty of Information Science
Hiroshima City University
Hiroshima 731-3194, Japan
miyazaki@hiroshima-cu.ac.jp

Takuro Namiki[1]
Faculty of Information Science
Hiroshima City University
Hiroshima 731-3194, Japan
e20144@e.hiroshima-cu.ac.jp

***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 in chromaticity space. In order to exaggerate the neighboring pixels as much as possible, we set the initial value randomly for iterative updation. On the other hand, in order to preserve the naturalness of the detailed structure, we weight parameters adequately to the cost function. Our method fully enhances the color difference between different color region so that the dichromats recognizes the color difference between neighboring pixels.**

***Index Terms*—color blindness, color vision deficiency, dichromat, recoloring, edge exaggeration**

I. Introduction

Enhancing the visibility of color image for dichromats is an important research field [1]–[13]. Most methods first map all pixel colors in color space such as RGB, HSV, XYZ, LMS, L*a*b*, etc., and next, they deform the color space or deform the clusters of mapped points so that it satisfies the required condition. On the other hand, our method analyzes the color difference between neighboring pixels. Namely, our method analyzes not in color space but in image space (*i.e.*, pixel coordinates). Also, unlike Chen [14], we do not preserve the color difference between two pixels but we increase the color difference as much as possible. We formulate the Poisson equation so that the relative color difference between neighboring pixels will be preserved.

Some methods [15]–[17] also solve the Poisson equation to enhance the visibility of dichromats. These methods [15], [16] form the Poisson equation in RGB intensity space, while our method forms in xy-chromaticity space. Existing method [17] also forms the Poisson equation in xy-choromaticity space. However, unlike [17], we set random values for initial values of iteration process in order to exaggerate the color difference between neighboring pixels as much as possible. On the other hand, our cost function preserves the detailed color difference between neighboring pixels. However, we multiply a weight parameter to each term of cost function for each pixel so that the color difference becomes large at the boundary where there is a large color gap.

[1]Currently: SCANET Inc., Tokyo, Japan

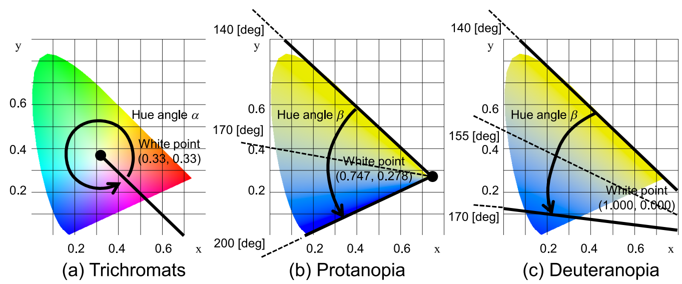

Fig. 1. Definition of hue of (a) trichromats, (b) protanopia, and (c) deuteranopia.

II. Color enhancement for dichromat

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. 1 (a)). The trichromatic hue α is defined as an angle around the white point $(x, y) = (0.33, 0.33)$.

The white point (the center of color confusion) of protanopia is $(x, y) = (0.747, 0.253)$ and that of deuteranopia is $(x, y) = (1.000, 0.000)$ [18]. The hue β is defined as the angle around these white points [6].

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

$$\mathbf{a}(u, v) = \begin{pmatrix} \tilde{x}(u, v) - 0.33 \\ \tilde{y}(u, v) - 0.33 \\ 0 \end{pmatrix} . \qquad (1)$$

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. (2).

$$\tilde{\mathbf{a}}(u+\Delta u, v+\Delta v) = \begin{pmatrix} \tilde{x}(u + \Delta u, v + \Delta v) - 0.33 \\ \tilde{y}(u + \Delta u, v + \Delta v) - 0.33 \\ 0 \end{pmatrix} . \quad (2)$$

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

$$\mathbf{a}(u+\Delta u, v+\Delta v) = \hat{\mathbf{a}}(u+\Delta u, v+\Delta v) \times \hat{\mathbf{a}}(u,v)\,. \quad (3)$$

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 $\Delta\alpha(u+\Delta u, v+\Delta v)$.

$$\Delta\alpha(u+\Delta u, v+\Delta v) = \sin^{-1}(\mathbf{a}(u+\Delta u, v+\Delta v))\,. \quad (4)$$

The difference of hue angle β between neighboring pixels should be proportional to the difference of hue angle α between neighboring pixels. Namely, the Laplacian of β should be the same as the Laplacian of α, scaled with a certain constant value.

$$\triangle\beta(u,v) = \triangle\alpha(u,v)\,. \quad (5)$$

Eq. (5) is called Poisson equation. The discretized representation of Eq. (5) is represented as follows.

$$\begin{aligned}&\beta(u,v) - \frac{1}{4}\beta(u-1,v) - \frac{1}{4}\beta(u+1,v)\\ &-\frac{1}{4}\beta(u,v-1) - \frac{1}{4}\beta(u,v+1) =\\ &\frac{1}{4}\Delta\alpha(u-1,v) + \frac{1}{4}\Delta\alpha(u+1,v)\\ &+\frac{1}{4}\Delta\alpha(u,v-1) + \frac{1}{4}\Delta\alpha(u,v+1)\,. \quad (6)\end{aligned}$$

The initial value of the hue angle β^0 is set to be a random number between -10 and 10. The number of iterations is denoted as k. By setting the initial value to a random number, the initial values of adjacent pixels become different colors. At the final stage, the colors of different color regions become as different as possible.

$$\beta^k(u,v) = w_1(u,v)\beta_1^k(u,v) + (1-w_1(u,v))\beta_2^k(u,v). \quad (7)$$

$$\beta_1^k(u,v) = \beta^{k-1}(u,v)\,. \quad (8)$$

$$\begin{aligned}&\beta_2^k(u,v) = \frac{1}{4}\{\\ &w_2(u,v;u-1,v)(\beta^{k-1}(u-1,v) + \Delta\alpha(u,v;u-1,v))\\ &+w_2(u,v;u+1,v)(\beta^{k-1}(u+1,v) + \Delta\alpha(u,v;u+1,v))\\ &+w_2(u,v;u,v-1)(\beta^{k-1}(u,v-1) + \Delta\alpha(u,v;u,v-1))\\ &+w_2(u,v;u,v+1)(\beta^{k-1}(u,v+1) + \Delta\alpha(u,v;u,v+1))\\ &+(1-w_2(u,v;u-1,v))\beta_3^k(u,v;u-1,v)\\ &+(1-w_2(u,v;u+1,v))\beta_3^k(u,v;u+1,v)\\ &+(1-w_2(u,v;u,v-1))\beta_3^k(u,v;u,v-1)\\ &+(1-w_2(u,v;u,v+1))\beta_3^k(u,v;u,v+1)\}\,. \quad (9)\end{aligned}$$

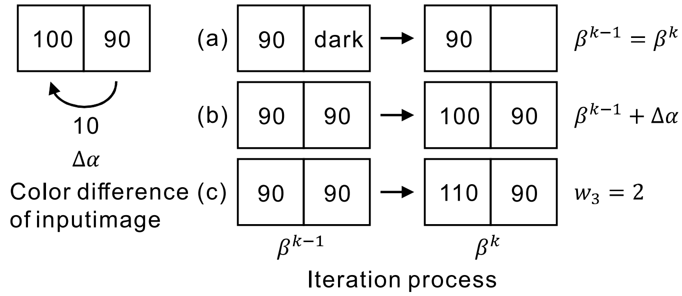

Fig. 2. Example of the calculation: (a) Ignoring dark pixels, (b) recovering the color difference, and (c) exaggerating the color difference.

$$\begin{aligned}&\beta_3^k(u,v;u-1,v) = \beta^{k-1}(u-1,v)\\ &+w_3(k)(\beta^{k-1}(u,v) - \beta^{k-1}(u-1,v))\,. \quad (10)\\ &\beta_3^k(u,v;u+1,v) = \beta^{k-1}(u+1,v)\\ &+w_3(k)(\beta^{k-1}(u,v) - \beta^{k-1}(u+1,v))\,. \quad (11)\\ &\beta_3^k(u,v;u,v-1) = \beta^{k-1}(u,v-1)\\ &+w_3(k)(\beta^{k-1}(u,v) - \beta^{k-1}(u,v-1))\,. \quad (12)\\ &\beta_3^k(u,v;u,v+1) = \beta^{k-1}(u,v+1)\\ &+w_3(k)(\beta^{k-1}(u,v) - \beta^{k-1}(u,v+1))\,. \quad (13)\end{aligned}$$

Each term of this equation is explained below.

Eq. (8) is used for pixels whose hue remain unchanged. If the pixel brightness is dark, the hue value is not reliable and should not be updated. Therefore, we use a weight to Eq. (7) that is calculated from the image brightness. As usual, we calculate the brightness I of the pixel (u,v) using Eq. (14) from RGB value.

$$I(u,v) = 0.299R(u,v) + 0.587G(u,v) + 0.114B(u,v)\,. \quad (14)$$

Weight w_1 is used to ignore dark pixels. Eq. (15) calculates the weight w_1 from the parameter m_1. An example of this effect is shown in Fig. 2 (a).

$$w_1(u,v) = e^{-m_1(I(u,v))^2}\,. \quad (15)$$

The darker the pixel is, the larger w_1 is, and the brighter the pixel is, the smaller w_1 is.

Eq. (9) is the Poisson equation (Eq. (5)) used to preserve the color difference between neighboring pixels. The weight w_2 is used to increase the large color changes, while leaving subtle color changes. Eq. (16) calculates the weight w_2 from the parameter m_2. The larger w_2 is, the more the original color difference is preserved.

$$w_2(u,v) = e^{-m_2(\Delta\alpha(u,v))^2}\,. \quad (16)$$

An example of the color difference between a pixel (u,v) and its right neighbor is shown in the Fig. 2 (b).

In the early stage of iteration, the large color changes are made larger, while in the later stage of iteration, the subtle color changes are preserved (Eq. (10)–(13)). Eq. (17) calculates the weight w_3 from the parameter m_3. Here, m_3

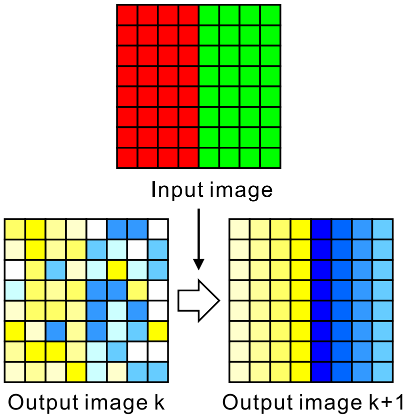

Fig. 3. Example of the calculation that reduce the noise and exaggerate the color boundary.

must be greater than 1 and close to 1. An example of this effect is shown in Fig. 2 (c).

$$w_3(k) = m_3^{k_{\max}-k} \,. \tag{17}$$

Here, the maximum number of iterations is denoted as $k_{\max}$. At the beginning of the iteration, w_3 is increased in order to change the hue angle β faster. However, if the hue angle changes too much, it diverges. Once the color changes to some extent, the amount of change is suppressed to stabilize the behavior. Therefore, w_3 is reduced for each iteration. Hue angle will not be changed when $w_3 = 1$. If $w_3 > 1$, the current color difference becomes w_3 times larger. When w_3 is large, the convergence speed becomes high since the hue angle changes drastically. When w_3 is small, the change of the hue angle becomes small, and the hue angle converges. This is because $\beta^{k-1}(u,v) - \beta^{k-1}(u+\Delta u, v+\Delta v)$ is the color difference in the current loop.

The weights w_1, w_2, w_3 calculated from the parameters m_1, m_2, m_3 affect the final result. An example of the behavior of the proposed method is shown in Fig. 3. We iterate the computation (Eq. (7)) for enough number.

III. Experiment

We applied our method to Fig. 4 (a). Fig. 4 (b) shows that the left doubled square is recognized as yellow and the right doubled square is recognized as blue. As is shown in Fig. 4 (c), our method uses both yellow and blue for both left and right squares. Both inner square and outer square are represented by different color so that dicrhomats can distinguish the boundary of each square.

Fig. 5 shows the result of our method applied to natural image. Input image (Fig. 5 (a)) includes red and green leaves, which are difficult to distinguish (Fig. 5 (b)). We iteratively (Fig. 5 (c)(d)(e)) update the hue. Final result (Fig. 5 (e)) shows that the same region is colored with same color, while the different regions are colored with different color.

Our results have higher color difference that use yellow and blue as much as possible. Our method has high performance

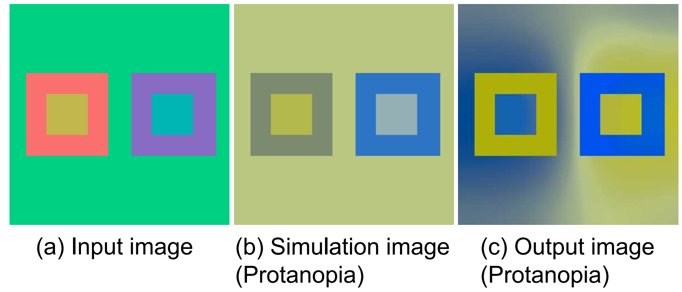

Fig. 4. Result of the proposed method applied to artificially generated image: (a) Input image, (b) protanopia view of input image, and (c) protanopia view of output image.

to exaggerate the color difference locally. On the other hand, our method paints the same color region with the same color.

IV. 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. Since we set the initial value as random numbers, the color difference between neighboring pixels become as large as possible. Our results show that the image is represented by blue and yellow so that the color difference is stretched as much as possible. Also, our results show that the local region with same input color becomes a region with same output color, which means that our method preserves the naturalness of the image locally.

The disadvantage of our method is the parameter tuning problem, so our future work is to find better cost function.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

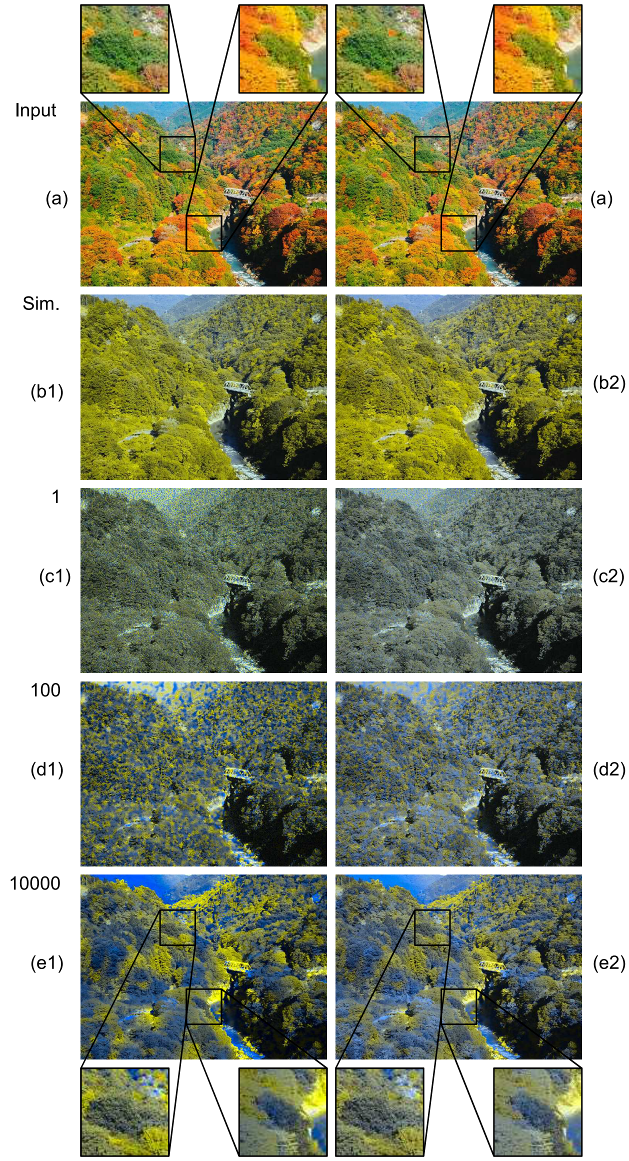

Fig. 5. Result of the proposed method applied to natural image: (a) Input image, (b) dichromats' image, (c) initial result of our method ($k = 1$), (d) intermediate result of our method ($k = 100$), and (e) final result of our method ($k = 10000$); (1) protanopia view and (2) deuteranopia view.

[8] M. M. Oliveira, "Towards more accessible visualizations for color-vision-deficient indificuals," *Computing in Science & Engineering*, vol. 15, no. 5, pp. 80–87, 2013.

[9] 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.

[10] 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.

[11] X. Zhang, M. Zhang, L. Zhang, P. Shen, G. Zhu, and P. Li, "Recoloring image for color vision deficiency by GANs," in *IEEE International Conference on Image Processing*, pp. 3267–3271, 2019.

[12] Z. Zhu, M. Toyoura, K. Go, I. Fujishiro, K. Kashiwagi, and X. Mao, "Processing images for red-green dichromats compensation via naturalness and information-preservation considered recoloring," *The Visual Computer*, vol. 35, pp. 1053–1066, 2019.

[13] Z. Zhu, M. Toyoura, K. Go, K. Kashiwagi, I. Fujishiro, T. Wong, and X. Mao, "Personalized image recoloring for color vision deficiency compensation," *IEEE Transactions on Multimedia*, vol. 24, pp. 1721–1734, 2022.

[14] L. Chen, Z. Zhu, W. Huang, K. Go, X. Chen, and X. Mao, "Image recoloring for color vision deficiency compensation using Swin transformer," *Neural Computing and Applications*, 16 pages, 2024.

[15] I. Farup, "Individualised halo-free gradient-domain color image daltonisation," *Journal of Imaging*, vol. 6, no. 116, 10 pages, 2020.

[16] 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.

[17] D. Miyazaki and S. Fujimura, "Hue enhancement for dichromats using Poisson equation," *Journal of Imaging Science and Technology*, fast track articles, 2021.

[18] D. B. Judd, "Standard response functions for protanopic and deuteranopic vision," *J. Optical Society of America*, vol. 35, pp. 199–221, 1945.