Luminance-Preserving and Temporally Stable Daltonization

P. Ebelin[1], C. Crassin[1], G. Denes[2,3], M. Oskarsson[4], K. Åström[4], and T. Akenine-Möller[1]

[1]NVIDIA [2]The Perse School Cambridge [3]University of Cambridge [4]Centre for Mathematical Sciences, Lund University

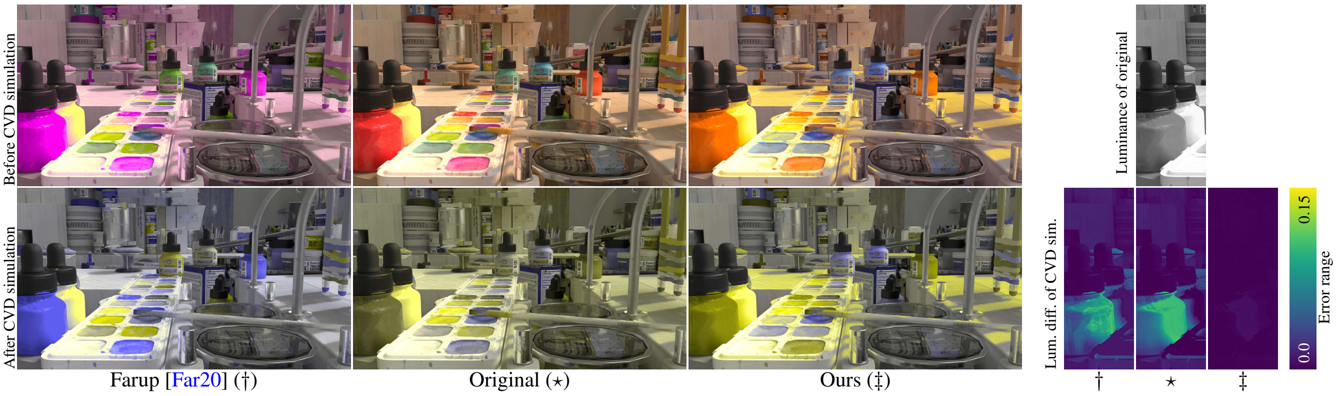

Figure 1: *Middle: original image (top) and the image as perceived by a person with a color vision deficiency (CVD; bottom). Our images (right) are similar to the original, improves contrast, and preserves luminance, where Farup's method (left) does not achieve the latter.*

Abstract

We propose a novel, real-time algorithm for recoloring images to improve the experience for a color vision deficient observer. The output is temporally stable and preserves luminance, the most important visual cue. It runs in 0.2 ms per frame on a GPU.

CCS Concepts

• ***Computing methodologies*** → ***Image processing; Rendering;***

1. Introduction

Color vision deficiencies (CVDs), more commonly known as color blindness, are often caused by genetics and affect the cones on the retina. Approximately 4.5% of the world's population (8% of males) has some form of CVD. Since it is hard for people with CVD to distinguish between certain colors, there might be a severe loss of information when presenting them with images as, e.g., reds and greens may be indistinguishable. To that end, several image-recoloring, *daltonization*, methods have been proposed, which aim to improve the experience for people with CVD [RG20, ZM21].

In this paper, we target a severe type of CVD called *dichromacy*, where an entire class of cone photopigment is missing [VBM99]. A dichromat may see only about 0.4% of the 16 million colors displayable with a 24-bit monitor, which makes the task of improving images using daltonization a challenging task. In fact, trying to present the dichromat with the same experience as someone with normal color vision is impossible.

Color confusion is unavoidable; however, achromatic acuity is known to be significantly higher than chromatic acuity, a phenomenon used by chroma subsampling in image compression algorithms. Thus, a good starting point for preserving image appearance is preserving luminance between the original image as seen by a person with normal color vision and someone with a CVD. Our focus is on generating and presenting images in real time, which includes rendering and video. Many previous algorithms attempt to daltonize images by exploiting the content of the image itself. While this is a suitable idea for an image in isolation, it often gives rise to temporal inconsistencies (e.g., flickering) when applied to a stream of images, as a color **c** could be mapped to a color **a** in one frame and **b** in another [RG20]. We present a novel daltonization algorithm that is temporally stable, fast, and luminance-preserving.

2. Previous Work

The field of recoloring for color vision deficiencies is vast, and we refer the reader to two recent surveys on this topic [RG20, ZM21].

DOI: 10.2312/egs.20231011

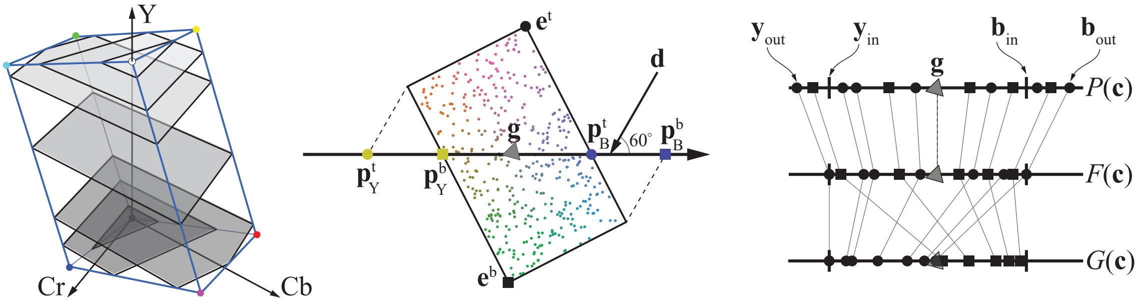

Figure 2: *Left: the RGB cube has been transformed into a parallelepiped in blue color using the YCbCr transform. A luminance polygon is the result of intersecting the parallelepiped with a fixed luminance. Seven different luminance polygons, at* $Y = \{0.05, 0.15, 0.25, 0.5, 075, 0.85, 0.95\}$*, are shown. Middle: the luminance polygon at* $Y = 0.5$ *with our notation, where the projection direction* **d** *(arrow), in this example, is angled 60° relative to the dichromacy line. Right: different linear remapping functions applied in our algorithm, where circles indicate that the original colors were located above the dichromacy line, and squares that they were located below.*

Machado and Oliveira [MO10] propose a real-time daltonization algorithm for dichromats. Their work focuses mainly on scientific visualizations and attempts to maximize contrast. While succeeding in maintaining temporal coherence in their examples, we found this not to be true in general (see Section 5). The technique considers the direction of maximum contrast loss, which is estimated based on a set of pixels chosen randomly at initialization. However, since the content of those pixels can change over time, so does the direction of maximum contrast loss and, consequently, the recoloring. Huang et al. [HWC08] propose to perform histogram equalization on the image's hues. As those hues may change during animation, this algorithm is also not temporally stable [RG20].

Farup [Far20] proposes a diffusion model for daltonization. While the main algorithm does not run in real time, his method for choosing the initial value of the optimization does, and we therefore compare to that. His technique is similar to Machado and Oliveira's. However, Farup modifies only the *difference* between the image as seen by someone with normal color vision and that as seen by a dichromat. The effect of this is improved naturalness preservation. Additionally, Farup's algorithm achieves temporal stability. A recent, real-time approach, which also considers differences in appearance, is that by Huang et al. [HZC*22]. However, their method also is not temporally stable (Section 5).

3. Background

There are many different types and severities of CVDs. While a daltonization algorithm should ideally handle each one, our algorithm mainly targets the severe case of *dichromacy*. For the dichromat, this means that the three-dimensional RGB gamut becomes a two-dimensional plane. We use the term *CVD simulation* for the process of simulating what a person with a certain type and severity of CVD experiences when viewing an image. There are several methods for this [MOF09, VBM99]. Importantly, it has been shown that people with a CVD perceive achromatic colors similar to people without a CVD [Jud48], implying that simulations should preserve grays.

Since we focus on dichromacy, in particular the more common types of protanopia and deuteranopia, we use the simulation method developed for those cases by Viénot et al. [VBM99]. If we assume an input, linearized sRGB color $\mathbf{c}$, and denote the CVD simulation function S, then the simulation method by Viénot et al. can be expressed as a multiplication of the color with a singular matrix, $\mathbf{M}_S$, i.e., $S(\mathbf{c}) = \mathbf{M}_S\mathbf{c}$ (see our supplemental PDF). We leverage the property of singularity when we decide the final output of our algorithm in Section 4.3. As mentioned earlier, the colors are contained in a plane after simulation. We call the intersection between this plane and an equi-luminant plane the *dichromacy line* (see Figure 2 center). We also refer to the subset of the line that is within the visible gamut by the *dichromat's line of visibility*.

4. Algorithm

Our daltonization method works per pixel (assuming that pixels contain linear sRGB values). On a high level it works as follows:

1. convert the input color $\mathbf{c}_i$ to a luminance-based color space as $\mathbf{c}$,
2. use a function T to map $\mathbf{c}$ to the dichromat's line of visibility,
3. back-project $T(\mathbf{c})$ to obtain $\mathbf{b}$, such that $S(\mathbf{b}) = T(\mathbf{c})$, and
4. finally, transform from linear color $\mathbf{b}$ to sRGB and display.

Next, we describe the steps of our method in detail.

4.1. Color Transform

As mentioned in the introduction, our algorithm is designed to preserve luminance. Hence, to facilitate development, we choose a color space that separates luminance and chrominance. For the remainder of this paper, we use a linear version of the YCbCr color space [Poy03]. YCbCr is the transform used in Rec. 709 for HDTV. Our algorithm requires a linear color space, which yields linear luminance in the Y-channel. Perceived brightness may be better represented by luma and we hope to explore this in future work. To the left in Figure 2, we show the vertices of the RGB cube transformed into YCbCr, which generates a parallelepiped.

4.2. Transformation to the Dichromat's Line of Visibility

This step aims to transform a color $\mathbf{c}$ to the dichromat's line of visibility. For this purpose, we use a projection direction $\mathbf{d}$ and simply project $\mathbf{c}$ along $\mathbf{d}$ (the arrow in Figure 2) until it intersects the dichromacy line. The projected point is denoted $P(\mathbf{c})$. The points above the dichromacy line are denoted *top* points and the ones below are called *bottom* points. In this case, the top points correspond to reds while the bottom points correspond to greens. The interval for the top points projected on the dichromacy line is $[\mathbf{p}_Y^t, \mathbf{p}_B^t]$, where Y indicates yellow and B blue, and t is for top. Similarly, the interval for the bottom is $[\mathbf{p}_Y^b, \mathbf{p}_B^b]$, where b is for bottom. For the inner pair of projected end points, we set $\mathbf{y}_{\text{in}}$ as the rightmost point of $\mathbf{p}_Y^t$ and $\mathbf{p}_Y^b$ and $\mathbf{b}_{\text{in}}$ as the leftmost point of $\mathbf{p}_B^t$ and $\mathbf{p}_B^b$. We use $\mathbf{y}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ for the outer pair of projected endpoints. See Figure 2. The projection direction $\mathbf{d}$ needs to be sufficiently different from the direction orthogonal to the dichromacy line in order to ensure enough separation between top and bottom points. In practice, using the direction shown in Figure 2 works well.

After projection, $P(\mathbf{c})$ may lay outside $[\mathbf{y}_{\text{in}}, \mathbf{b}_{\text{in}}]$. We therefore define a linear remapping function $Q(\mathbf{p}, \mathbf{p}_s, \mathbf{p}_e, \mathbf{q}_s, \mathbf{q}_e)$, which assumes that $\mathbf{p}$ lies on the line segment from $\mathbf{p}_s$ to $\mathbf{p}_e$ and remaps $\mathbf{p} = \mathbf{p}_s + t(\mathbf{p}_e - \mathbf{p}_s)$ to $\mathbf{q}_s + t(\mathbf{q}_e - \mathbf{q}_s)$. In order to preserve the gray point $\mathbf{g}$, we remap the projected colors as follows. If the projected color $P(\mathbf{c})$ is to the left of the gray point, $\mathbf{g}$, on the dichromacy line, then the remapped point is $Q(P(\mathbf{c}), \mathbf{y}_{\text{out}}, \mathbf{g}, \mathbf{y}_{\text{in}}, \mathbf{g})$. Similarly, if the projected color $P(\mathbf{c})$ is to the right of the gray point, we remap it as $Q(P(\mathbf{c}), \mathbf{g}, \mathbf{b}_{\text{out}}, \mathbf{g}, \mathbf{b}_{\text{in}})$. The resulting color after this remapping is denoted $F(\mathbf{c})$. See the top step to the right in Figure 2.

We want the final projected points to preserve colors $\mathbf{c}$ close to the dichromacy line, while making it possible to distinguish between red and green colors further away from the line. To make that possible, we introduce a differentiating mapping G, which maps points above the dichromacy line to the left of the gray point and points below to the right of it. For points above, this is accomplished by $Q(F(\mathbf{c}), \mathbf{y}_{\text{in}}, \mathbf{b}_{\text{in}}, \mathbf{y}_{\text{in}}, \mathbf{g})$, and for points below, by $Q(F(\mathbf{c}), \mathbf{y}_{\text{in}}, \mathbf{b}_{\text{in}}, \mathbf{g}, \mathbf{b}_{\text{in}})$. This step is shown at the bottom right in Figure 2. The final projected point is an interpolated intermediary color $H(\mathbf{c}) = (1-t)F(\mathbf{c}) + tG(\mathbf{c})$, where t is proportional to the ratio of the height of $\mathbf{c}$ from the dichromacy line and the maximum height of the top or bottom side of the luminance polygon, depending on whether $\mathbf{c}$ is situated on the top or bottom side, respectively. In practice, we use $t = \min(3r, 1)$, where r is the mentioned ratio, and the factor 3 was found empirically.

To increase the usage of the available colors on the dichromat's line of visibility, the final step performs a *separate* weighted histogram equalization of colors $H(\mathbf{c})$ to the left of the gray point and to the right of it. Since $H(\mathbf{c})$ is only dependent on $\mathbf{c}$'s position inside the luminance polygon, we precompute histograms of the colors that H transforms to the right and left sides of the gray point by uniform sampling of the entire polygon. Equalization is done by computing a cumulative distribution function from the histogram, and accessing it for all the points $H(\mathbf{c})$. We use linear interpolation between two adjacent bins during access, which assumes (as a reasonable approximation) a linear relation between them.

While the above gave good results for the mid-range of luminances, it could give skewed results in the darker and brighter areas. For example, a dark gray shadow could become dark blue. To counteract this, we weight the histogram counts with a function $w(i, l)$, where i is the bin index starting from the gray point and going outward on both sides, and l is the luminance, i.e.,

$$w(i,l) = (0.1 + 0.9s)(1-q) + q, \tag{1}$$

where $q = i/(n-1) \in [0,1]$ for n bins and $s = 1 - 2|l - 0.5|$. This weights the bin counts with a linear ramp, starting at $w = 0.1$ for $i = 0$, at low and high luminances and w goes toward a constant function, $w = 1$, for $l = 0.5$. Applying histogram equalization of $H(\mathbf{c})$, using the weighted histograms, gives us the final color $T(\mathbf{c})$ on the dichromat's line of visibility.

4.3. Back-Projection

A dichromat will perceive several of the RGB colors as the same color under the assumption that their visible gamut is two-dimensional [VBM99]. We leverage this to determine the final, displayed color. To make our output more accessible to people without a CVD, or to one with a lower severity, we aim to present an image that is as similar to the original image as possible, under the constraint that the dichromat should perceive the color $T(\mathbf{c})$ that we computed earlier. Our approach is to present a back-projected color $\mathbf{b}$ which satisfies $S(\mathbf{b}) = T(\mathbf{c})$. Because the CVD simulation matrix, $\mathbf{M}_S$, is a pure projection [VBM99], it is singular and its nullspace consists of a single vector, $\mathbf{n}$. Due to the properties of the nullspace, we have $S(T(\mathbf{c}) + t\mathbf{n}) = T(\mathbf{c})$ for all $t \in \mathbb{R}$, i.e., all colors $T(\mathbf{c}) + t\mathbf{n}$ are perceived as $T(\mathbf{c})$ by the dichromat. We compute $\mathbf{b}$ using $\mathbf{n}$ as

$$\mathbf{b} = T(\mathbf{c}) + \big(\mathbf{n} \cdot (\mathbf{c} - T(\mathbf{c}))\big)\mathbf{n}. \tag{2}$$

This gives us the color closest to the original color, $\mathbf{c}$, while also being perceived as $T(\mathbf{c})$ by the dichromat. Should the point $\mathbf{b}$ fall outside the color space's visible gamut, we convert it to RGB and clamp to $[0,1]^3$. This may cause a small change in luminance compared to the original image. Several methods were evaluated but their overall results were not better than those from clamping.

5. Evaluation

Many previous daltonization approaches do not provide temporal stability. From one frame to the next, a significantly different recoloring may be produced. This is the case for the algorithm by Huang et al. [HZC*22] and by Machado and Oliveira [MO10] (see our supplemental material). Since temporal stability is integral in real-time rendering, we do not compare further to these algorithms.

We consider the image results in Figures 1 and 3. We simulate protanopia in the former and deuteranopia in the latter. The images on the right in the figures show that our algorithm improves the perceived luminance for the dichromat. In the top row of Figure 1, we see that our backprojection technique increases the similarity between our output and the original image compared to Farup's algorithm. The awnings in Figure 3 indicate the same. However, as the backprojection is constrained by the colors we want the dichromat to perceive, i.e., $T(\mathbf{c})$, its output cannot always be set that close to the original. See, for example, the tree in the bottom left. Our supplemental PDF contains additional image results as well as details about the performance of our algorithm.

To quantitatively compare daltonization algorithms, we consider their performance on color vision tests. Such tests are commonly used to diagnose people with CVD. As daltonization algo-

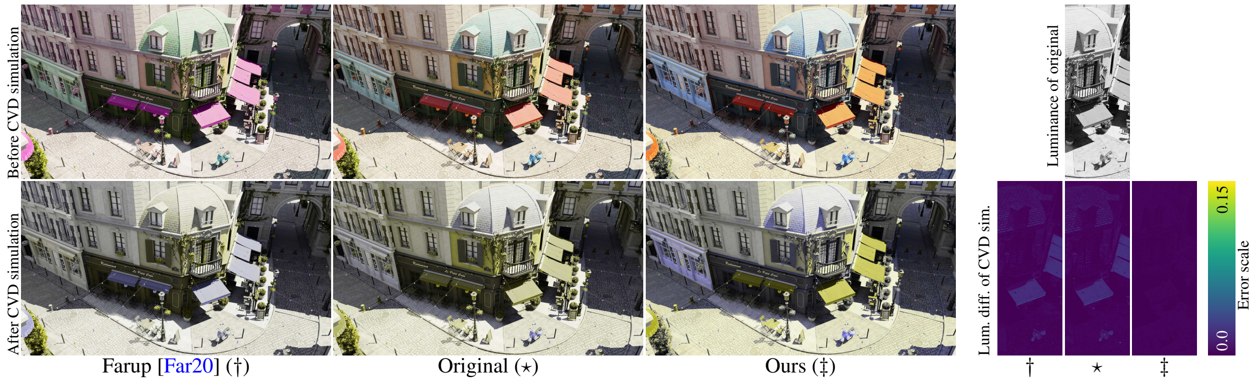

Figure 3: *This figure has the same layout as Figure 1. Note the luminance loss for Farup's alg. and CVD sim. of the original image.*

Table 1: *TES on the simulated CBFM test (lower is better). Simulated CVD performance on our proposed daltonization algorithm is comparable to no daltonization, but it is noticeably better than the simulated performance of Farup [Far20].*

	Orig.	Orig.+CVD	Our+CVD	Farup+CVD
TES	75.3	170.1	181.2	231.2

rithms aim to improve the experience of CVD observers, we postulate that observer performance on these tests after daltonization of the stimuli could provide valuable insights. The Farnsworth-Munsell 100-hue test [Far43] and its computer-based equivalent (CBFM) [GPD*14] quantify visual performance through the conceptually simple task of arranging colored tiles (caps) according to hues. An equi-luminant hue circle is split into four sections (boxes), with the first and last color of each section fixed. The total error score (TES) can be computed as the sum of absolute differences between cap locations and adjacent cap locations [Far43]. There are variations in how TES is computed, and the score can change for the same person. However, generally TES greater than 100 can be considered an indication of CVD.

To compare our daltonization algorithm with Farup's, we simulate the results of the CBFM tests. To each color cap we apply either Farup's or our daltonization algorithm and then simulate CVD using Viénot et al. [VBM99]. To simulate the behaviour of a human observer, we then examine the ΔE_{00} color distance between caps and apply a stochastic process to order the tiles. The algorithm is calibrated to correspond to TES scores for non-CVD and CVD people reported in the literature [GPD*14]. More details on the simulation can be found the supplemental material. Our results (Table 1) indicate that our proposed daltonization does not yield improvements over no daltonization for the CBFM experiment, but it performs consistently better than Farup's [Far20].

6. Conclusions and Future Work

Our method is temporally stable by design, runs in real time on a GPU, and retains luminance substantially better than the competition. In future work, we want to include a user study and augment our algorithm to handle lower CVD severities.

Acknowledgments We thank ©Amazon Lumberyard for the BISTRO scene. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[Far43] FARNSWORTH D.: The Farnsworth-Munsell 100-Hue and Dichotomous Tests for Color Vision. *Journal of the Optical Society of America 33*, 10 (Oct 1943), 568–578. 4

[Far20] FARUP I.: Individualised Halo-Free Gradient-Domain Colour Image Daltonisation. *Journal of Imaging 6*, 11 (2020). 1, 2, 4

[GPD*14] GHOSE S., PARMAR T., DADA T., VANATHI M., SHARMA S.: A New Computer-Based Farnsworth Munsell 100-Hue Test for Evaluation of Color Vision. *International Ophthalmology 34*, 4 (2014), 747–751. 4

[HWC08] HUANG J.-B., WU S.-Y., CHEN C.-S.: Enhancing Color Representation for the Color Vision Impaired. In *Workshop on Computer Vision Applications for the Visually Impaired* (2008). 2

[HZC*22] HUANG W., ZHU Z., CHEN L., GO K., CHEN X., MAO X.: Image Recoloring for Red-Green Dichromats with Compensation Range-based Naturalness Preservation and Refined Dichromacy Gamut. *The Visual Computer 38* (2022), 3405–3418. 2, 3

[Jud48] JUDD D. B.: Color Perceptions of Deuteranopic and Protanopic Observers. *Journal of Research of the National Bureau of Standards USA 41* (1948), 247–271. 2

[MO10] MACHADO G. M., OLIVEIRA M. M.: Real-Time Temporal-Coherent Color Contrast Enhancement for Dichromats. *Computer Graphics Forum 29*, 3 (2010), 933–942. 2, 3

[MOF09] MACHADO G. M., OLIVEIRA M. M., FERNANDES L. A. F.: A Physiologically-based Model for Simulation of Color Vision Deficiency. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1291–1298. 2

[Poy03] POYNTON C.: *Digital Video and HDTV – Algorithms and Interfaces*. Morgan Kaufmann, 2003. 2

[RG20] RIBEIRO M., GOMES A. J. P.: Recoloring Algorithms for Colorblind People: A Survey. *ACM Computing Surveys 52*, 4 (2020), 1–37. 1, 2

[VBM99] VIÉNOT F., BRETTEL H., MOLLON J. D.: Digital Video Colourmaps for Checking the Legibility of Displays by Dichromats. *Color Research & Application 24*, 4 (1999), 243–252. 1, 2, 3, 4

[ZM21] ZHU Z., MAO X.: Image Recoloring for color Vision Deficiency Compensation: a Survey. *The Visual Computer 37* (2021), 2999–3018. 1