Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

Real-Time Temporal-Coherent Color Contrast Enhancement for Dichromats

Gustavo M. Machado and Manuel M. Oliveira

Instituto de Informática - UFRGS, Brazil

Abstract

We present an automatic image-recoloring technique for enhancing color contrast for dichromats whose computational cost varies linearly with the number of input pixels. Our approach can be efficiently implemented on GPUs, and we show that for typical image sizes it is up to two orders of magnitude faster than the current state-of-the-art technique. Unlike previous approaches, ours preserve temporal coherence and, therefore, is suitable for video recoloring. We demonstrate the effectiveness of our technique by integrating it into a visualization system and showing, for the first time, real-time high-quality recolored visualizations for dichromats.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Introduction

According to current estimates, approximately 200 million individuals present some form of color vision deficiency (CVD) [SSJN99, Rig99]. Such a condition compromises the ability of these individuals to effectively perform color and visualization-related tasks, affecting both their private and professional lives [OH88]. This situation is particular restrictive for *dichromats* and *monochromats*. A dichromat is an individual that misses one of the three types of photopigments required for normal color vision in humans. Dichromats can be classified according to the kind of the missing photopiment as *protanopes*, *deuteranopes*, or *tritanopes*, if the missing photopigment is more sensitive to the long, medium, or short wavelengths of the visible spectrum, respectively. Monochromats, on the other hand, have only one or no type of photopigment, and are classified as *cone monochromats* or *rod monochromats*, respectively.

The recent popularization of digital imaging technology and color displays has provided an opportunity for minimizing the loss of color contrast experienced by individuals with CVD. This has prompted the interest of several researchers to the problem of image recoloring for dichromats [ITK*04, JH06, RGW05a, RGW05b, WS05, KOF08a]. None of these techniques, however, is sufficiently fast to provide real-time high-quality image recoloring, or preserve temporal coherence. As a result, they are not suitable for use in real-time or even in interactive visualization systems.

We present a real-time technique for enhancing color contrast for dichromats that guarantees temporal coherence and preserves achromatic colors (gray shades). The cost of our approach is linear on the number of input pixels, and most of the computation can be performed independently for each pixel, lending to an efficient GPU implementation. We demonstrate the effectiveness of our approach by using it to obtain real-time, temporal-coherent, high-quality visualizations for dichromats.

Figure 1 illustrates some results generated by our technique and compares them with the ones produced by the state-of-the-art technique for image recoloring for dichromats [KOF08a]. The reference images represent the perception of normal trichromats. The column *Dichromat* shows the simulated perceptions of dichromats for the corresponding reference images. The next two columns show the recolored images obtained using our technique and its exaggerated contrast version, respectively. The two remaining columns show the results produced by the regular and by the exaggerated-contrast versions of the recoloring technique of Kuhn et al. [KOF08a]. Note how our technique can satisfactorily recover the contrast lost by dichromats.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

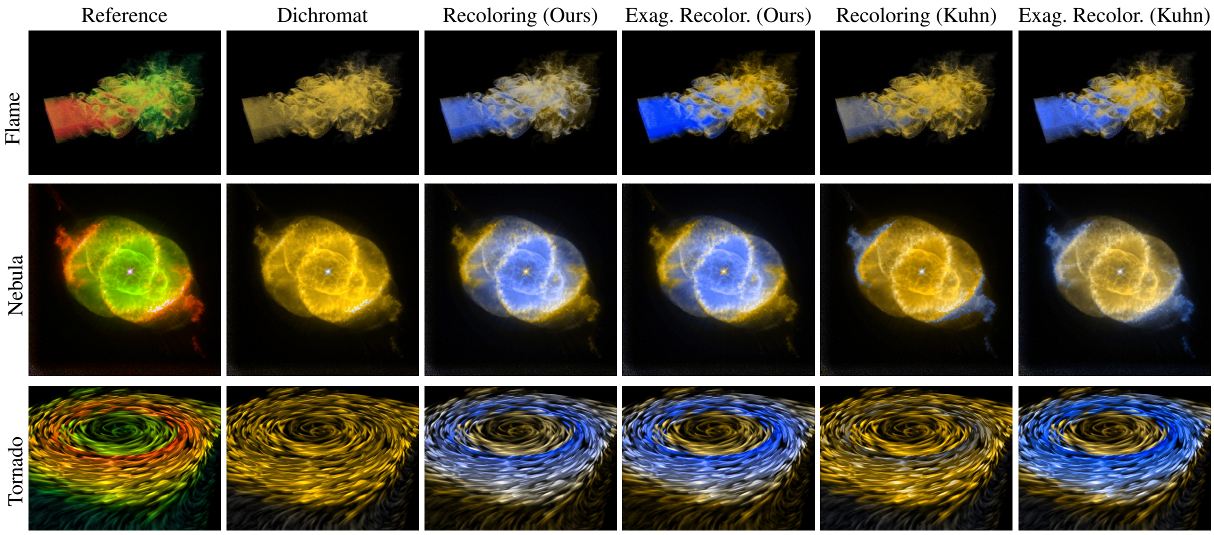


Figure 1: *Comparison of the results produced by our recoloring technique and by Kuhn et al.'s [KOF08a] for a set of scientific visualization images. The "Dichromat" column shows the simulated perception of dichromats for the corresponding "Reference" image obtained using the approach of Machado et al. [MOF09]. The simulation and recolorings of the Flame and Nebula images are for deuteranopes, while the Tornado ones are for protanopes.*

The main contributions of this paper include:

- The first contrast-enhancement image-recoloring technique for dichromats that produces high-quality results in real time (Section 3). Our solution scales well with the number of input pixels, can be efficiently implemented on GPUs, and preserves achromatic colors;
- A technique that enforces temporal coherence in the recolored image sequences (Section 3.3);
- The first demonstration of a visualization application with support for real-time high-quality image recoloring for dichromats (Section 4);

2. Related Work

Several researchers have investigated the problem of image-recoloring for individuals with CVD. The existing techniques can be broadly classified as *user-assisted* and *optimization-based* approaches.

2.1. User-Assisted Techniques

The techniques in this class require assistance, in the form of user-provided parameters, to guide the recoloring process. Thus, the quality of their results is highly dependent on the provided parameters, making them unsuitable for real-time systems. Iaccarino et al. [IMPS06] employ six parameters to modulate the original colors of an input image. Daltonize [DW02] uses three parameters to specify the recoloring process (for protanopes and deuteranopes). These parameters specify how the red-green channel should be stretched, projected into the luminance channel, and projected into the yellow-blue channel.

2.2. Optimization-based Techniques

These techniques operate without user intervention and consist of optimization procedures. Ichikawa et al. [ITK*03] used a genetic algorithm to recolor web pages for anomalous trichromats. Subsequently, the authors extended their work for image recoloring [ITK*04]. Wakita and Shimamura [WS05] presented a technique for recoloring documents (*e.g.*, web pages, charts, maps) for dichromats. Such a technique is based on three objective functions intended for: (i) color contrast preservation, (ii) maximum color contrast enforcement, and (iii) color naturalness preservation (for user-specified colors). The three objective functions are weighted according to user-specified parameters and optimized with simulated annealing. Wakita and Shimamura report that the optimization for documents with more than 10 colors could take several seconds. Jefferson and Harvey [JH06] use four objective functions to preserve brightness, color contrast, colors in the available gamut, and color naturalness. Their technique optimizes the combined objective functions using preconditioned conjugate gradients. The authors reported times of the order of several minutes for a set of 25 key colors (on a P4 2.0 GHz PC using Matlab).

Rasche et al. [RGW05a] presented a recoloring technique for dichromats consisting of an optimization that tries to preserve the perceptual color differences between all pairs of colors using an affine transformation. However, such a transformation does not capture color variations along many directions and does not guarantee that the mapped colors are constrained to the available gamut. In a subsequent work, Rasche et al. [RGW05b] addressed these limitations applying a constrained multivariate optimization procedure to a reduced set of quantized colors. The resulting set of optimized quantized colors is then used to optimize the entire set of colors. Despite the improved results, this algorithm is prone to local minima, and does not scale well with the number of quantized colors and the size of the input images.

Kuhn et al. [KOF08a] presented a technique for enhancing color contrast for dichromats based on mass-spring optimization, which can be efficiently implemented on GPUs. Similar to the technique of Rasche et al. [RGW05b], the optimization is first performed on a set of quantized colors, which are then used to optimize the entire set of colors. Although their technique is about three orders of magnitude faster than previous approaches and can achieve interactive frame rates, it is still not sufficiently fast to allow real-time performance. Moreover, since the optimization is based on a set of quantized colors, it is not clear how one could preserve temporal coherence on the fly (*e.g.*, during an interactive scientific visualization session).

2.3. Color-to-Grayscale Mappings

Image recoloring for dichromats is a dimensionality reduction problem. In this sense, it is akin to the more constrained problem of color-to-grayscale mapping. Traditional techniques commonly used in commercial applications [Bro06, Jes02] perform this mapping by simply taking the color's luminance value computed on some color space (*e.g.*, *XYZ*, *YCbCr*, *L*a*b**, or *HSL*). An important aspect of all these techniques is that they preserve achromatic colors, which is a desirable feature for printing. Since no chrominance information is taken into account, these approaches map all isoluminant colors to the same shade of gray, despite of their perceptual differences. Recently, several techniques have been proposed to address this limitation [GOTG05, RGW05b, GD07, KOF08b].

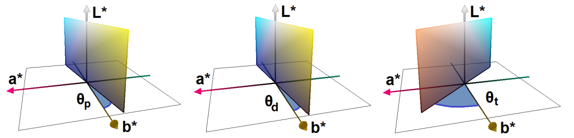
Gooch et al. [GOTG05] use an optimization procedure whose cost is quadratic in the number of pixels in the image. Although the technique produces some good results, its computational cost precludes it from being used for interactive applications. Moreover, it does not preserve achromatic colors. Gooch et al. report that they have explored the use of principal component analysis (PCA) to estimate an ellipsoid in color space that best approximates the set of colors found in the image. The grayscale image would then be computed by projecting all image colors on the axis of the ellipsoid with the largest variance. According to the authors [GOTG05] and also pointed out by Rasche et al. [RGW05b], PCA fails to convert color images with variations along many directions, and an optimization step would be required to somehow combine the principal components.

Grundland and Dogdson [GD07] perform the color-to-grayscale mapping by adding to the original luminance value Y_i of pixel p_i, some amount K_i that tries to compensate for the contrast loss. To compute K_i while avoiding a quadratic cost (as in previous techniques), the authors introduced a clever local sampling strategy called *Gaussian pairing*. It consists in choosing, for each pixel p_i, a pixel p_j in a circular neighborhood around p_i. The choice of p_j is based on a Gaussian probability distribution function. The size of the neighborhood is computed based on the image dimensions. For a given pair (p_i, p_j), the relative contrast loss is computed as:

$$l_{(p_i,p_j)} = 1 - \frac{Y_i - Y_j}{\|p_i - p_j\|_{RGB}}, \qquad (1)$$

where Y_i and Y_j are the luminance values of pixels p_i and p_j, respectively, and $\|p_i - p_j\|_{RGB}$ is the length of the color vector $v_{ij} = p_i - p_j$ computed in the *RGB* color space. Note that the distance computed in the denominator of Equation 1 has no perceptual meaning. In order to estimate the amount K_i, the authors map the original RGB colors to their own opponent-color space (*YPQ*), which, again, is not perceptually uniform. In the YPQ color space, they estimate a direction d_{mcl} of maximum contrast loss (according to Equation 1) using a technique of their own, which they called *predominant component analysis*. The idea of predominant component analysis is to approximate the direction of maximum data dispersion using a sum of weighted vectors. Note, however, that its results are not equivalent to the solution of an eigenvector problem, such as done in PCA. As a result, for the same set of input vectors, the direction d_{mcl} is not the same as the direction of the main eigenvector obtained using PCA. For computing K_i, the authors essentially project the original pixel colors expressed in the YPQ space onto d_{mcl}.

Our approach uses the Gaussian pairing technique of Grundland and Dogdson [GD07] for acceleration. However, ours differs from their approach in many fundamental aspects. First, we are dealing with recoloring for dichromats, as opposed to color-to-grayscale mappings. Second, we perform all the computation in the $L^*a^*b^*$ color space, which is an approximately perceptually uniform color space, where it makes sense to use distances to represent perceptual differences. Third, we use PCA, an established technique for estimating the direction of maximum variance in a given dataset, to compute the direction v_{ab} that maximizes the loss of color contrast (in a least-squares sense) in the chromaticity plane (*i.e.*, a^*b^* plane). Fourth, we use the coordinates of the projected colors onto the plane defined by v_{ab} and the L^* axis as the transformed color coordinates, as opposed to using them to complement the original luminance values. And finally, our approach presents temporal coherence, which is not supported by the technique described in [GD07].



(a) (b) (c)

Figure 2: *Planar approximation for the color gamut of dichromats in the CIE $L^*a^*b^*$ color space [KOF08a]. (a) Protanope ($\theta_p = -11.48°$). (b) Deuteranope ($\theta_d = -8.11°$). (c) Tritanope ($\theta_t = 46.37°$).*

3. The Color-Contrast Enhancing Technique

Our approach is based on the key observation that, whenever dichromats experience some significant loss of color contrast, most of this contrast can be recovered by working on a perceptually uniform color space, and orthographically projecting the original colors onto a plane aligned with the direction that maximizes contrast loss (in a least-squares sense). The coordinates of these projections then become the new color coordinates on the reduced (2D) color gamut of the dichromat. Figure 3 summarizes this process, which consists of the following steps:

1. Estimation of the vector v_{ab} that represents the direction that maximizes contrast loss in the CIE $L^*a^*b^*$ chromaticity plane, and
2. Projection of the original colors onto the plane defined by v_{ab} and the lightness (L^*) axis. The projected color coordinates are then rotated around L^* to align themselves to the plane of the dichromat, and the resulting colors are used to recolor the image.

In order to guarantee temporal coherence, we check and correct for abrupt changes in the sense of v_{ab}. Sections 3.1 to 3.3 present the details of these steps.

3.1. Direction that Maximizes Contrast Loss

The color gamut of each class of dichromats can be represented by two half-planes in the LMS color space [BVM97], and can be satisfactorily approximated by a single plane [VBM99]. Figure 2 shows these planes mapped to the CIE $L^*a^*b^*$ color space. According to Kuhn et al. [KOF08a], the angles between such planes and the L^*b^* plane are $\theta_p = -11.48°$, $\theta_d = -8.11°$, and $\theta_t = 46.37°$, for protanopes, deuteranopes, and tritanopes, respectively. The colors shown in Figure 2 represent the actual color gamut for each class of dichromacy.

Computing the direction that maximizes the loss of local contrast (in the least-squares sense) for a dichromat observing an image I would require evaluating, for each pixel $p_i \in I$, the contrast lost between p_i and all pixels p_j in a neighborhood N_i around p_i. However, due to spatial coherence, neighbor pixels tend to have similar values. Thus, estimating the loss of local contrast for each pixel p_i against all pixels in N_i tends to result in a significant amount of redundant work. We avoid performing this computation on the entire neighborhood N_i by resorting to the Gaussian pairing sampling technique of Grundland and Dogdson [GD07]. In this case, for each pixel p_i, its contribution to the loss of local contrast is estimated from a single neighbor p_j. The horizontal and vertical distances between p_i and p_j are randomly defined by univariate Gaussian distributions with zero mean and variance $(2/\pi)\sigma^2$, where $\sigma^2 = \sqrt{2min(width, height)}$ [GD07], the function $min(a,b)$ returns the minimum of a and b, and *width* and *height* are the dimensions of the image. Although the direction of maximum loss of local color contrast obtained using this sampling strategy differs from the one computed using the entire neighborhoods, due to spatial coherence these directions tend to be sufficiently close to each other. To avoid flickering during the recoloring of animated or video sequences (Section 3.3), we pre-compute the coordinates of the p_js and store them in a texture. The same pairs (p_i, p_j) are then used during the entire sequence.

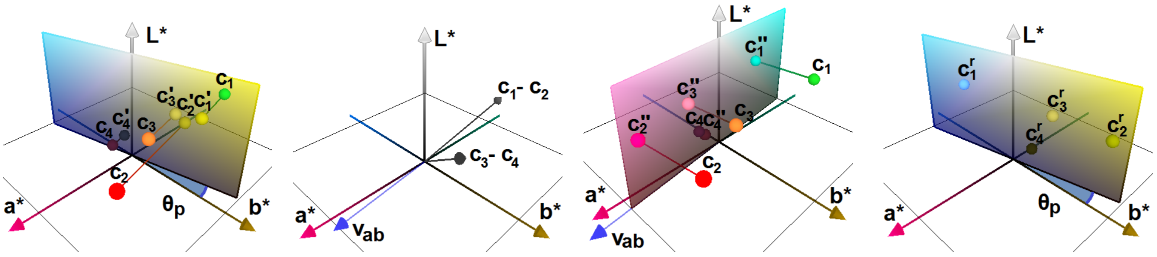
Figures 3 (a) and (b) illustrate the process of computing the direction that maximizes the loss of contrast for two pairs of colors, (c_1, c_2) and (c_3, c_4), represented as small spheres in the $L^*a^*b^*$ color space. c'_1 to c'_4 are the projections of colors c_1 to c_4, respectively, on the dichromat's plane, and represent his/her perception of colors c_1 to c_4 (Figure 3(a)). Since L*a*b* is approximately perceptually uniform, the relative loss of color contrast experienced by a dichromat observing a pair of colors (c_i, c_j) (with respect to a normal color vision observer) can be estimated as

$$l_{(c_i,c_j)} = \frac{\|c_i - c_j\| - \|c'_i - c'_j\|}{\|c_i - c_j\|}, \tag{2}$$

where $\|.\|$ is the vector length operator. For this pair of colors, the direction of contrast loss is given by $\vartheta_{ij} = c_i - c_j$. Since we preserve the lightness coordinate (*i.e.*, L^*) of the original colors to avoid polarity reversal [KOF08a], it suffices to compute the direction that maximizes contrast loss on the chromaticity plane. Computing it in the entire $L^*a^*b^*$ space does not improve the results, and would require finding the eigenvectors of a 3×3 matrix instead of a 2×2. Thus, let $w_i = l_{(c_i,c_j)}\vartheta_{ij}$ be the vector representing the contrast loss along direction ϑ_{ij} associated to pixel p_i. Also let

$$M = \begin{bmatrix} w_1^{a^*} & w_1^{b^*} \\ w_2^{a^*} & w_2^{b^*} \\ \dots & \dots \\ w_n^{a^*} & w_n^{b^*} \end{bmatrix} \tag{3}$$

be a matrix whose rows contain the coordinates of the chromaticity vectors representing the loss of color contrast for all pixels in image I. The elements of the i-th row of M are the projections of w_i on the a^*b^* plane. v_{ab} can then be obtained as the eigenvector of the 2×2 matrix $M^T M$ whose corresponding eigenvalue has the largest absolute value. In



(a) (b) (c) (d)

Figure 3: *The steps of our recoloring algorithm. (a) Colors c_1 to c_4 are perceived by a dichromat as c'_1 to c'_4, respectively (their projections on the dichromat's gamut plane). The relative loss of contrast experienced by a dichromat for a pair of colors (c_i, c_j) is given by $l_{(c_i,c_j)} = (\|c_i - c_j\| - \|c'_i - c'_j\|)/(\|c_i - c_j\|)$, which happens along the direction $\vartheta_{ij} = c_i - c_j$. (b) Direction v_{ab} (shown in blue) that maximizes the loss of local contrast (in a least-square sense) is computed as the main eigenvector of the matrix $M^T M$, where M is defined in Equation 3. (c) Projection of the original colors on the plane defined by v_{ab} and L^*. (d) Final colors obtained after rotating the projected colors c''_k in (c) around L^* so that they align with the dichromat's plane.*

order to solve the resulting characteristic equation, we set the b^* coordinate of v_{ab} to one and solve for its a^* coordinate.

3.2. Computing the Final Colors

When a dichromat experiences a significant loss of color contrast, the orthographic projection of the original colors onto the plane defined by the vectors L^* and v_{ab} tends to spread these colors more than when they are projected onto the dichromat's plane. This situation is illustrated on Figures 3 (a) and (c). If the spatial relationship among the projected colors onto the L^*-v_{ab} plane is transfered to the dichromat's plane, an image with better color contrast will be produced. This is achieved by rotating the projected colors around the L^* axis so that they now align with the plane of the dichromat. Figure 3 (d) illustrates this operation, which preserves the achromatic colors from the original image.

Exaggerated Contrast: As one maps the *RGB* cube to the $L^*a^*b^*$ color space, the maximum length of the resulting chromaticity vectors is 148.47. Thus, we obtain images with exaggerated contrast, similar to the ones produced by Kuhn et al.'s technique, simply by rescaling all chromaticity coordinates in the recolored image so that its maximum chroma becomes 148. We emphasize, however, that this is not the preferred use for our technique, as images produced with exaggerated contrast tend to contain higher perceptual distortions than images obtained using our regular recoloring technique. Recolorings with exaggerated contrast are shown in Figures 1, 4 and 8 for the sake of comparison.

3.3. Enforcing Temporal Coherence

Temporal coherence is an important requirement for image-recoloring techniques, as subtle changes in the color of an object during an interactive visualization session or animation can be quite disturbing. This section explains how our technique enforces temporal coherence.

If the direction of v_{ab} is almost parallel to the a^* axis, a minor change in the input colors might imply a switch in the sign of v_{ab}'s small $b*$ component (see Figure 3(b)). However, for the solution of the characteristic equation associated with the computation of v_{ab}, we have arbitrated its b^* component to 1 (Section 3.1), as one possible way to avoid the trivial solution $v_{ab} = 0$. This constrains v_{ab}'s b^* component to be positive, causing its a^* component to change its sign to accommodate the change in v_{ab}'s direction. Such a change would cause recolored pixels to abruptly change colors between consecutive frames (*e.g.*, blue pixels would turn yellow, and vice versa). We avoid the occurrence of these artifacts during an interactive visualization session (or animation) by saving the vector v_{ab} of the current frame and using it for comparison in the next frame. As the angle between the previous and current vectors approaches 180°, we invert the sense of the current v_{ab} to enforce color consistency. Although simple, this is an effective solution, whose results can be observed in the accompanying video.

4. Results

We have implemented the described technique both for CPU and for GPU, using C++ and GLSL, and used them to recolor a large number of images. We have also integrated the GPU implementation with an existing visualization application using a minimally invasive approach. In this session, we compare the performance and quality of the results produced by our technique with the ones obtained with the approach of Kuhn et al. [KOF08a], which is currently the state-of-the-art both in terms of performance and image quality. One should note that all recolored images shown in the paper are perceived similarly both by the class of dichromats they were recolored for and by normal color vision individuals. This

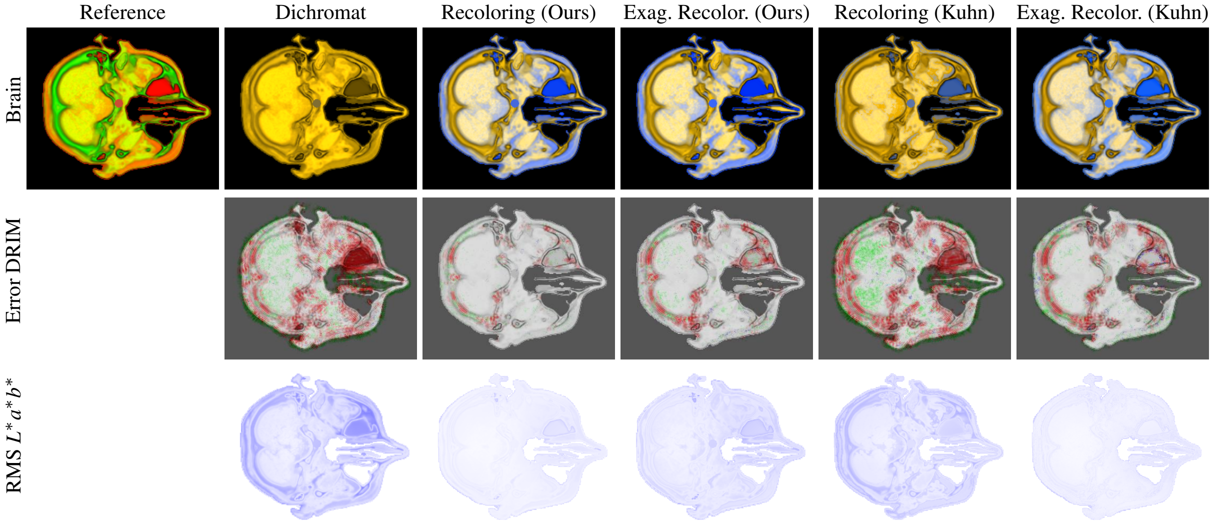


Figure 4: *Comparison of the results produced by our technique and by Kuhn et al.'s. First row, from left to right: reference image, simulated perception of a deuteranope, and recolored images using various algorithms. Second row: Perceptual errors according to the DRIM metric. Third row: Local contrast differences according to the RMS metric of Equation 4 (for k* $= 100$*). According to both metrics, our recoloring technique is less prone to noticeable changes in contrast.*

comes from the fact that all colors used for recoloring are on the dichromat's plane, which is a subset of the color gamut of a normal trichromat. Also, all images exemplifying results of Kuhn et al.'s techniques (regular and exaggerated contrast) were generated with a CPU implementation based on K-means (with up to 128 clusters) in order to obtain the best possible image quality. The results reported in the paper were obtained using a Core 2 Extreme 3.0 GHz PC with 8 GB of memory and a Quadro FX 5800 graphics card.

We assess the quality of our results using both subjective comparison and a perceptual image quality metric. For this, we use the dynamic range independent image quality metric (DRIM) of Aydin et al. [AMMS08]. DRIM uses a model of the human visual system to try to detect visible changes in image structure. Figures 1, 4, and 8 show results produced by our technique and compare them with the ones obtained with Kuhn et al.'s approach. In these figures, the reference images illustrate the perception of normal trichromats. The column *Dichromat* shows the simulated perception of dichromats obtained with the algorithm of Machado et al. [MOF09]. The remaining columns present the recolored images obtained using our technique, an exaggerated contrast version of it, Kuhn et al.'s [KOF08a] technique, and its exaggerated contrast version, respectively. The reference images are from different sources. They illustrate cases of effective visualizations for normal trichromats, but which are challenging for individuals with CVD. As such, they provide good test cases for our technique. Most of these images (*Flame, Brain, Knee, Foot, Europe* and *Chart*) have also been used by Kuhn et al. [KOF08a]. Due to space constraints, Figure 1 includes recolorings, but not the error images produced by DRIM (they favor our results).

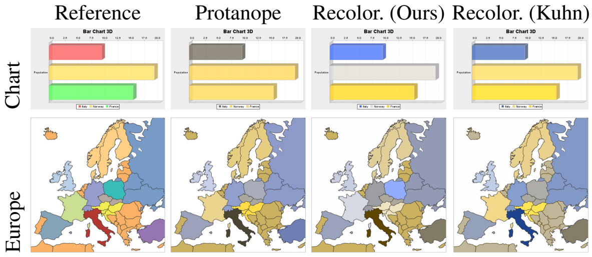


Figure 5: *Recoloring of information visualization images for protanopes.*

Figure 4 shows the results obtained when recoloring the reference *Brain* image for deuteranopes. The second row shows their corresponding color-coded perceptual errors according to DRIM (using its default parameters), computed with respect to the reference image. Green indicates *loss of contrast*, blue represents *contrast amplification*, and red shows regions with *contrast reversal*. The more saturated the colors, the higher the probability of a human observer perceiving these changes in contrast. The DRIM results suggest that the metric cannot fully capture contrast changes that result purely from image recoloring.

In order to complement DRIM's results, we defined a simple error metric that tries to capture local differences in color

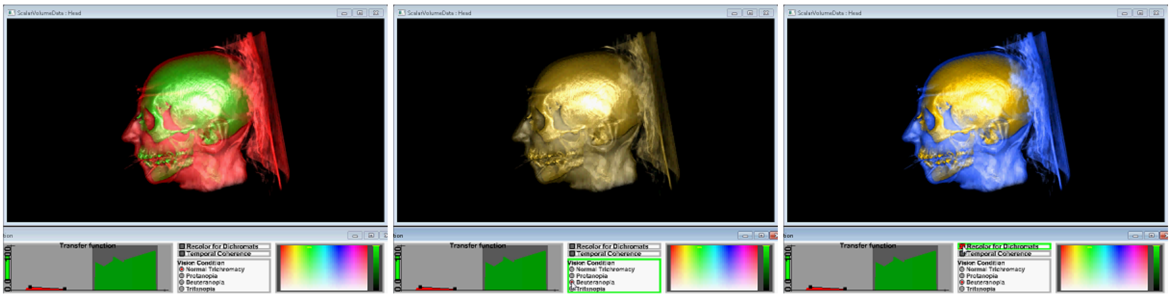


Figure 6: *Integration of our technique with an existing visualization application. (left) Reference image as perceived by an individual with normal color vision. (center) Simulation of the perception of a deuteranope for the reference image using the model of Machado et al. [MOF09]. (right) Recolored image for a deuteranope using our technique.*

contrast between pairs of images. Such a metric consists in computing a root-mean-square (RMS) error across corresponding neighborhoods in the (pair of) images using the $L^*a^*b^*$ color space. Although it produces plausible results, this metric has not been formally validated as a perceptual error metric. Due to space restrictions, we only present its results for the example shown in Figure 4. The proposed RMS metric is defined as follows: let $p_i \in I_{ref}$ be a pixel in the reference image I_{ref}, and let $S_{pi} = \{p_j, p_{j+1}, ..., p_{j+k-1}\}$ be the set of pixels also in I_{ref} in a neighborhood containing k pixels centered at p_i. Likewise, let $q_i \in I_{test}$ be the pixel corresponding to p_i in the test image I_{test}, and let $S_{qi} = \{q_j, q_{j+1}, ..., q_{j+k-1}\}$ be the set of pixels in the corresponding k-neighborhood around q_i. The difference of local color contrast between I_{ref} and I_{test} at p_i and q_i can be expressed as

$$RMS(q_i) = \sqrt{\frac{1}{k}\sum_{s=j}^{j+k-1}\left(\frac{(p_i - p_s) - (q_i - q_s)}{160}\right)^2}. \tag{4}$$

The constant 160 in the denominator keeps the resulting value in the $[0,1]$ range. The third row of Figure 4 shows the resulting error images obtained applying Equation 4 to the foreground pixels of the images in the first row (with respect to the reference one). The value 0 is shown in white, and darker shades of blue indicate bigger errors. Note how these errors vary more smoothly over the images. According to both metrics, the results produced by our technique are less prone to noticeable changes in contrast than the regular and exaggerated versions of Kuhn et al.'s technique.

Figure 8 compares our results and Kuhn et al.'s for three medical visualization images. Again, for these examples, our recoloring technique introduced less contrast distortions than both the regular and exaggerated versions of Kuhn et al.'s approach. Kuhn et al.'s technique [KOF08a] is based on the more sounding principle of estimating perceptual differences between pairs of colors in the reference image, and using an optimization procedure to try to enforce such differences in the recolored image. In practice, however, the optimization often might not be able to recover the optimal contrast due to the occurrence of local minima. This explains the results shown in Figures 1, 4, and 8. Our projection-based approach, on the other hand, tends to produce good results whenever a dichromat experiences significant loss of color contrast.

Figures 5 shows ours and Kuhn et al.'s results for two information visualization examples. In this case, only the results of the regular recoloring techniques are shown. Table 1 summarizes the performance of our approach in comparison to both the CPU and GPU versions of Kuhn et al.'s approach for seven images shown in the paper. The entries in the table are ordered by the number of pixels in the images. One can observe that both versions of our technique are up to two orders of magnitude faster than Kuhn et al.'s counterparts on images up to 800×800 pixels. Since the cost of our algorithm is linear on the number of pixels, the achieved speedup improves as the image size increases. Although Kuhn et al.'s mass-spring optimization applied to a set of quantized colors is quite efficient, it still requires a quantization and a reconstruction steps, which dominate the total cost of the algorithm. Table 2 summarizes these costs for some of the images shown in the paper, considering the two quantization and reconstruction strategies described by the authors.

Our technique can be easily integrated with existing applications using a minimally invasive approach. It can be deployed as a few shader programs, which the application should call after rasterizing the scene, but before swapping the back and front buffers. In this case, the shaders simply read the content of the back buffer, recolor it, and write it back, after which the host application swaps the buffers. We used this strategy to integrate our technique to an existing visualization application. Figure 6 shows some snapshots of the resulting system. The accompanying video shows the resulting application in action.

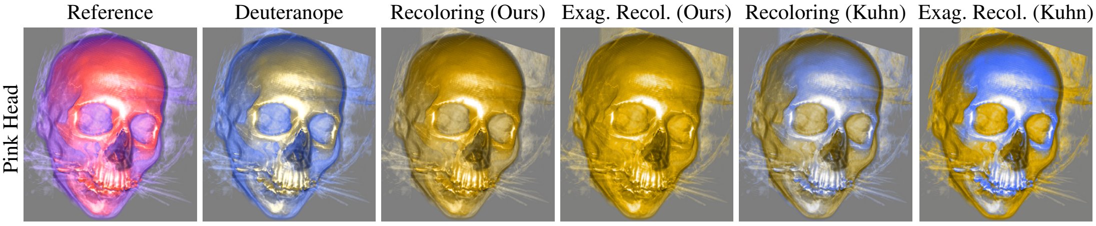


Figure 7: *Example of a situation that causes our technique to fail. Note that deuteranopes (and protanopes) already perceive the reference image as having sufficient contrast, and no recoloring is necessary.*

Image (size)	**CPU**		**GPU**	
	Our Time	**Kuhn Time**	**Our Time**	**Kuhn Time**
Flame (288x184)	0.055	1.148	0.019	0.121
Chart (500x300)	0.191	2.707	0.020	0.106
Foot (446x446)	0.252	3.743	0.021	0.307
Brain (532x523)	0.292	5.053	0.023	0.577
Knee (528x528)	0.341	5.217	0.022	0.313
Europe (596x486)	0.385	5.361	0.023	0.565
Nebula (800x800)	0.614	11.73	0.028	1.145

Table 1: *Performance comparison (in sec.) of our technique and Kuhn et al.'s [KOF08a] for both CPU and GPU versions of the algorithms executed on several images. Due to the linear cost of our approach, the relative speedup improves as the image size increases.*

Image (size)	**Clusters**	**Quant. time**	**Reconst. time**	
Flame (288x184)	88	0.9363	0.0027	**K-Means**
Foot (446x446)	78	3.5055	0.0082	
Europe (596x486)	70	5.1130	0.0115	
Nebula (800x800)	98	11.2448	0.0255	
Flame (288x184)	77	0.0155	0.0857	**Uniform**
Foot (446x446)	48	0.0670	0.2764	
Europe (596x486)	48	0.0978	0.5356	
Nebula (800x800)	74	0.2120	1.1130	

Table 2: *Times (in sec.) for the quantization and reconstruction phases of the technique of Kuhn et al. [KOF08a] for several images. K-means used in the CPU version of the technique, and uniform quantization used in its GPU version. The column Clusters shows the number of clusters identified in the quantization phase.*

4.1. Limitations

Like all previous image-recoloring techniques for dichromats, ours also has some limitations. For example, let d be the direction of maximum dispersion of the original colors in image I (in $L^*a^*b^*$). If v_{ab} is approximately perpendicular to d and the color dispersion along d is larger than along v_{ab}, the recolored image should exhibit less contrast than the original one. In practice, this requires that the loss of contrast be small, meaning that the dichromat could already perceive the details in the original image. In such a situation, there would be no need for recoloring in the first place. Figure 7 illustrates this situation with an image whose colors were carefully chosen to achieve this effect. According to our experience, such cases should happen only rarely. To handle them, the user can turn our recoloring technique *on* and *off* at any time during an interactive session.

Although our technique preserves gray shades, it does not provide a mechanism for preserving other colors perceived similarly by dichromats and normal color vision individuals, as does the technique described in [KOF08a]. Also, as a fundamental limitation of the reduced color gamut of dichromats, no technique, including ours, can fully recover the lost contrast in all situations.

5. Conclusion

We have presented the first technique to provide real-time, temporal-coherent, high-quality image recoloring for dichromats. Its computational cost varies linearly with the number of input pixels, and it can be efficiently implemented on GPUs. We have shown that the results produced by our technique are at least as good as the ones obtained with the current state-of-the-art technique, while being up to two orders of magnitude faster. We have also shown how to integrate our technique with existing applications using a minimally invasive strategy, and demonstrated its effectiveness producing real-time visualizations for dichromats.

Our results should enable the development of more user-friendly applications for individuals with color vision deficiency. For instance, the low computational cost of our technique makes it a suitable solution for implementation on cell phones and other mobile devices equipped with cameras. On such devices, the recoloring capabilities of our technique can be a useful tool for assisting color vision deficient individuals in several daily tasks.

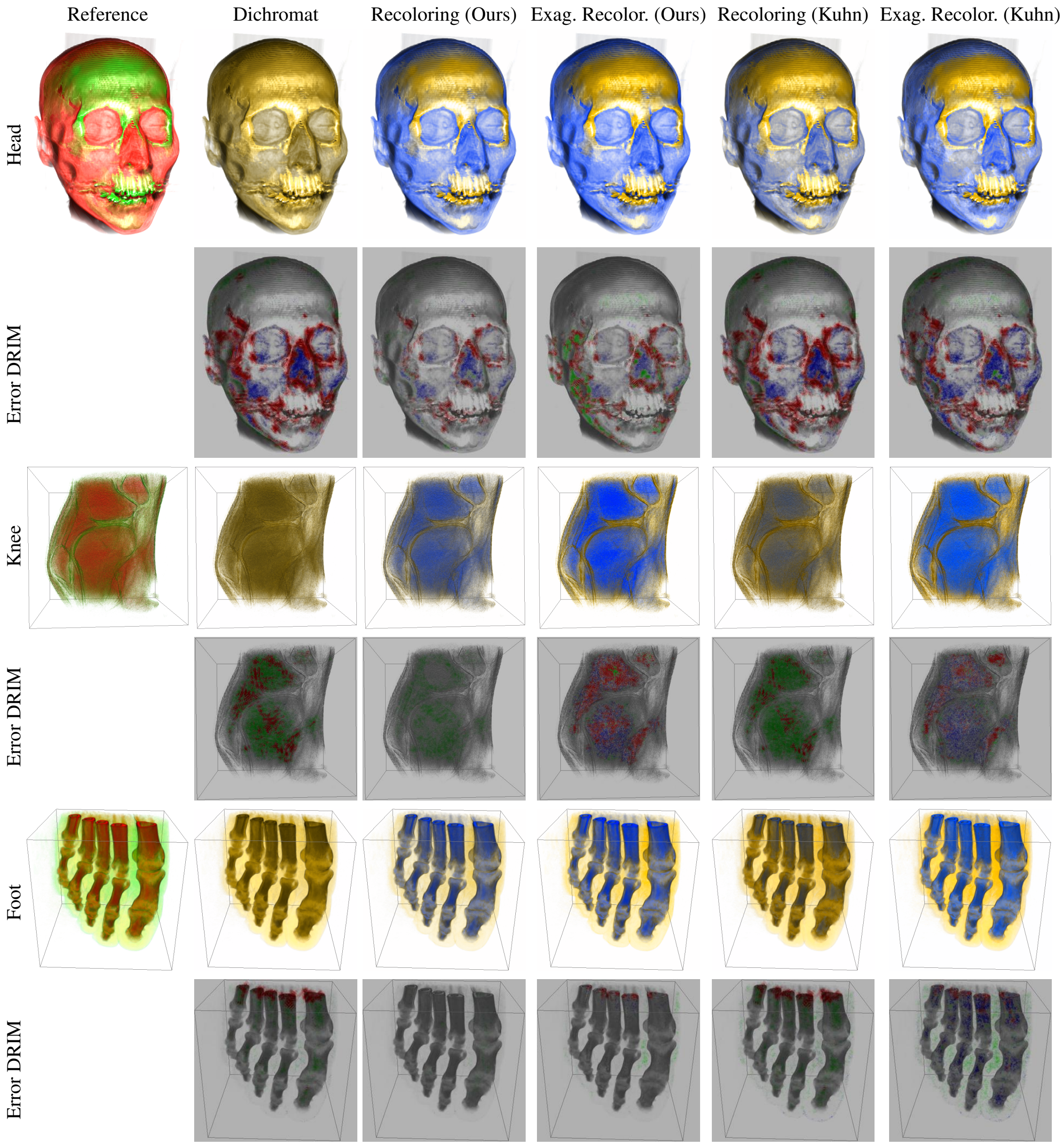


Figure 8: *Comparison of the results produced by our technique and the ones obtained with Kuhn et al.'s approach for a set of medical visualization images. The even rows show the estimated changes in contrast perceived by an observer in the recolored images (with respect to the reference images), according to the DRIM metric. Green indicates* loss of contrast*, blue represents* contrast amplification*, and red shows regions with* contrast reversal*. The metric favors our results in all examples.*

6. Acknowledgments

We thank Giovane Kuhn for many fruitful discussions and suggestions, and Francisco Pinto for providing the visualization software. We also thank the anonymous reviewers for their insightful comments and suggestions. This work was sponsored by CNPq-Brazil (grants 200284/2009-6, 131327/2008-9, 476954/2008-8, and 305613/2007-3). Nvidia kindly donated the Quadro FX 5800 card used in this research. The reference images were kindly provided by Francisco Pinto (Brain, Foot, and Knee), CCSE at LBNL (Flame), Martin Falk and Daniel Weiskopf (Tornado), and http://commons.wikimedia.org (Nebula, Chart, and Europe).

References

[AMMS08] AYDIN T. O., MANTIUK R., MYSZKOWSKI K., SEIDEL H.-P.: Dynamic range independent image quality assessment. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–10.

[Bro06] BROWN R.: Photoshop: Converting color to black-and-white. (http://www.russellbrown.com/tips_tech.html), 2006.

[BVM97] BRETTEL H., VIÉNOT F., MOLLON J. D.: Computerized simulation of color appearance for dichromats. *J. Opt. Soc. Am. 14*, 10 (1997), 2647–2655.

[DW02] DOUGHERTY R., WADE A.: Daltonize. (http://www.vischeck.com/daltonize), 2002. Accessed on Nov/09.

[GD07] GRUNDLAND M., DODGSON N. A.: Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recogn. 40*, 11 (2007), 2891–2896.

[GOTG05] GOOCH A. A., OLSEN S. C., TUMBLIN J., GOOCH B.: Color2gray: salience-preserving color removal. *ACM Trans. Graph. 24*, 3 (2005), 634–639.

[IMPS06] IACCARINO G., MALANDRINO D., PERCIO M. D., SCARANO V.: Efficient edge-services for colorblind users. In *Proc. of WWW '06* (2006), pp. 919–920.

[ITK*03] ICHIKAWA M., TANAKA K., KONDO S., HIROSHIMA K., ICHIKAWA K., TANABE S., FUKAMI K.: Web-page color modification for barrier-free color vision with genetic algorithm. *Lecture Notes in Computer Science 2724* (2003), 2134–2146.

[ITK*04] ICHIKAWA M., TANAKA K., KONDO S., HIROSHIMA K., ICHIKAWA K., TANABE S., FUKAMI K.: Preliminary study on color modification for still images to realize barrier-free color vision. In *IEEE SMC '06* (2004), vol. 1, pp. 36–41.

[Jes02] JESCHKE E. R.: Gimp: Converting color images to b&w. (http://www.gimp.org/tutorials/Color2BW/), 2002. Accessed in Aug/07.

[JH06] JEFFERSON L., HARVEY R.: Accommodating color blind computer users. In *Proc. of ASSETS '06* (2006), pp. 40–47.

[KOF08a] KUHN G. R., OLIVEIRA M. M., FERNANDES L. A. F.: An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE TVCG 14*, 6 (2008), 1747–1754.

[KOF08b] KUHN G. R., OLIVEIRA M. M., FERNANDES L. A. F.: An improved contrast enhancing approach for color-to-grayscale mappings. *Vis. Comput. 24*, 7 (2008), 505–514.

[MOF09] MACHADO G. M., OLIVEIRA M. M., FERNANDES L. A. F.: A physiologically-based model for simulation of color vision deficiency. *IEEE TVCG 15*, 6 (2009), 1291–1298.

[OH88] OLSEN M. M., HARRIS K. R.: *Color Vision Deficiency and Color Blindness: An Introduction to the Problem*. Fern Ridge Press, 1988.

[RGW05a] RASCHE K., GEIST R., WESTALL J.: Detail preserving reproduction of color images for monochromats and dichromats. *IEEE Comput. Graph. Appl. 25*, 3 (2005), 22–30.

[RGW05b] RASCHE K., GEIST R., WESTALL J.: Re-coloring images for gamuts of lower dimension. *Comput. Graph. Forum 24*, 3 (2005), 423–432.

[Rig99] RIGDEN C.: The eye of the beholder - designing for colour-blind users. *British Telecommunications Engineering 17* (1999).

[SSJN99] SHARPE L. T., STOCKMAN A., JÄGLE H., NATHANS J.: *Color Vision: From Genes to Perception*. Cambridge University Press, 1999, ch. Opsin genes, cone photopigments, color vision, and color blindness, pp. 3–51.

[VBM99] VIÉNOT F., BRETTEL H., MOLLON J. D.: Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research and Application 24* (1999), 243–252.

[WS05] WAKITA K., SHIMAMURA K.: Smartcolor: disambiguation framework for the colorblind. In *Proc. of ASSETS '05* (2005), pp. 158–165.