Methods for computing color anaglyphs

David F. McAllister[a], Ya Zhou[b], Sophia Sullivan[c]

[a]North Carolina State Univ., Dept. of Computer Science, Raleigh, North Carolina, USA 27695;
[b]Beijing Institute of Technology, Haidian District, Beijing 100081, P. R. China;
[c]Rose-Hulman Institute of Technology, Terre Haute, Indiana, USA 47803

ABSTRACT

A new computation technique is presented for calculating pixel colors in anaglyph images. The method depends upon knowing the RGB spectral distributions of the display device and the transmission functions of the filters in the viewing glasses. It requires the solution of a nonlinear least-squares program for each pixel in a stereo pair and is based on minimizing color distances in the CIEL*a*b* uniform color space. The method is compared with several techniques for computing anaglyphs including approximation in CIE space using the Euclidean and Uniform metrics, the Photoshop method and its variants, and a method proposed by Peter Wimmer. We also discuss the methods of desaturation and gamma correction for reducing retinal rivalry.

INTRODUCTION

Several methods have been proposed to compute anaglyphs [14,15], stereo images that require colored filters over each eye to view the image. The filters are used to block wavelengths produced by the display device so that each eye sees a different image. Blocking is based on the spectral distributions of the primaries in the display and the transmission functions of the filters. Perfect blocking would require filters that subdivide the visible wavelengths into two disjoint regions so that each eye received unique colors, thereby preventing ghosting or the mixing of left and right eye views. Ghosting, however, is a common problem in stereo viewing. For a discussion of ghosting in an anaglyph environment, see Andrew Woods, et. al. [01]. Another problem with anaglyphs is binocular or retinal rivalry. This phenomenon occurs when different colors are presented to each eye. Rivalry is distracting and contributes to visual fatigue and other strange side effects. Consequently, minimization of rivalry has also received considerable attention in practice. See for example [02] and [03].

In [04], McAllister, et. al. compare several common techniques for computing pixel colors in anaglyphs including the Photoshop method (abbreviated PS), the Modified Photoshop method (MPS), Dubois's Least Squares method [05] where the Euclidean norm is minimized in Commission Internationale de l'Eclairage (CIE) color space (LS), and the Uniform approximation method where the uniform or Chebychev norm is minimized in CIE space (UN). In [04], the UN was shown to be superior to the others in color representation. PS and MPS are not dependent on the characteristics of the filters or the properties of the display device. Wimmer [06] has proposed two methods for computing anaglyphs. We give examples of his improved technique in the appendix.

The CIELab method described in this paper is similar to LS and UN in that it minimizes a metric or norm in a color space, but in this case we use the Euclidean norm and move to the color space CIEL*a*b*. The color space is *uniform* where equal Euclidean distances represent perceptually equal color and luminance changes. Our method was developed to improve color representation in anaglyphs. In converting the RGB coordinates to CIEL*a*b* we first convert RGB to CIE. The matrices that are used depend on the properties of the display device and the anaglyph filters. A nonlinear transformation is applied to move from the CIE color space to CIEL*a*b*. Hence, the computation of a pixel color requires the minimization of a nonlinear least squares function, normally using an iterative process that can be computationally intensive. Since the CIEL*a*b* color space is nonlinear and distances are measured using the Euclidean norm, approximation using the uniform norm was not considered.

In Section 2 we review the conversion to CIEL*a*b* and the nonlinear least squares problem that results. In Section 3 we analyze the behavior of the method and compare it to UN. In Section 4 we consider methods for reducing rivalry. Section 5 contains a summary with conclusions and areas for future research. References are next, and anaglyph examples are at the end, in the Appendix.

2. THE CIELAB METHOD

In 1931, the Commission Internationale de l'Eclairage, in an attempt to create a standard color system, created the CIE XYZ space, also called the norm color system [07]. Colors in this space are a function of the tristimulus values X, Y, and Z, and are easily converted to and from RGB space using linear transformations (matrix multiplications) that depend on the properties of the spectral distributions of the RGB primaries of the display device. The transmission functions of the red/cyan anaglyph glasses and the RGB spectral decompositions for LCD displays used in this paper are given in [04].

The operations required to compute the matrix values for the filter and RGB spectral distributions of the display device are described in [04] and [05]. The latter is particularly important because RGB distributions can be very different, especially for projection displays, a particular inconvenience since most manufacturers will not share this information.

The matrix C used to convert RGB to CIE on a cathode ray tube, or CRT display system, can be found in [08].

On an LCD display, the C matrix is as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4243 & 0.3105 & 0.1657 \\ 0.2492 & 0.6419 & 0.1089 \\ 0.0265 & 0.1225 & 0.8614 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

We normalize X, Y, and Z by dividing each by the sum. Then we only need two coordinates to represent a color, $x = X/(X+Y+Z) \;\; and\; y = \; Y/(X+Y+Z)$. This is equivalent to slicing the color solid with a plane that intersects the points (1,0,0), (0,1,0), and (0,0,1).

CIEL*a*b* conversion involves CIE coordinates using cube root functions. The values Xn, Yn, and Zn are normalized tristimulus values that account for the reflectance of the object and the light shining on the body [09]. The values used in this research are (Xn, Yn, Zn) = (11.144, 100, 35.201). The CIEL*a*b* conversion equations are as follows [09]:

$$L^*[X,Y,Z] = 116\left(\frac{Y}{Yn}\right)^{\frac{1}{3}} - 16$$

$$a^*[X,Y,Z] = 500\left[\left(\frac{X}{Xn}\right)^{\frac{1}{3}} - \left(\frac{Y}{Yn}\right)^{\frac{1}{3}}\right]$$

$$b^*[X,Y,Z] = 200\left[\left(\frac{Y}{Yn}\right)^{\frac{1}{3}} - \left(\frac{Z}{Zn}\right)^{\frac{1}{3}}\right]$$

A linear spline f is used to compute L* for values below 0.008856: if $s \leq 0.008856, f(s) = 7.787s + \frac{16}{116}$. Except for small values, the formula for luminance is the same in conversion to the CIELuv color space, also a uniform space [18].

The object is then to find the RGB values red (*r*), green (*g*), and blue (*b*) when converted to CIEL*a*b* coordinates that minimize the Euclidean (least squares) distance to the CIEL*a*b* colors of the left eye pixel and the right eye pixel simultaneously. The matrices *Al* and *Ar* convert RGB values to CIE values when viewed through the filters. Let C_{left} denote the RGB values for a pixel in the left image and C_{right} the color for the same pixel in the right eye image. For LCD displays and the red/cyan glasses used in this research we have:

$$CIEleft\left[C_{left}\right] = Al\left(C_{left}\right) = \begin{bmatrix} 0.1840 & 0.0179 & 0.0048 \\ 0.0876 & 0.0118 & 0.0018 \\ 0.0005 & 0.0012 & 0.0159 \end{bmatrix} C_{left}$$

$$CIEright\left[C_{right}\right] = Ar\left(C_{right}\right) = \begin{bmatrix} 0.0153 & 0.1092 & 0.1171 \\ 0.0176 & 0.3088 & 0.0777 \\ 0.0201 & 0.1016 & 0.6546 \end{bmatrix} C_{right}$$

The above matrices are the same as those used in [04]. The problem is then to find *r*, *g*, and *b* to minimize the following objective function. As for LS and UN, the map is normalized so that the optimal solution for white (255, 255, 255) is white.

$$\begin{aligned}&\left(L^*[r,g,b]-L^*\left[CIEleft\left[C_{left}\right]\right]\right)^2+\left(a^*[r,g,b]-a^*\left[CIEleft\left[C_{left}\right]\right]\right)^2\\&+\left(b^*[r,g,b]-b^*\left[CIEleft\left[C_{left}\right]\right]\right)^2+\left(L^*[r,g,b]-L^*\left[CIEright\left[C_{right}\right]\right]\right)^2\\&+\left(a^*[r,g,b]-a^*\left[CIEright\left[C_{right}\right]\right]\right)^2+\left(b^*[r,g,b]-b^*\left[CIEright\left[C_{right}\right]\right]\right)^2\end{aligned}$$

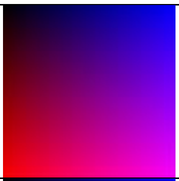
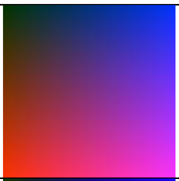
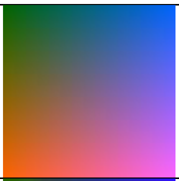
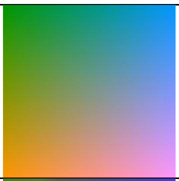
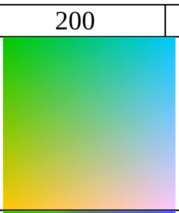
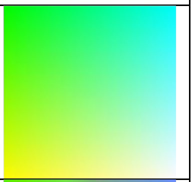
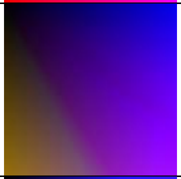
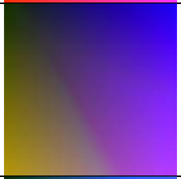
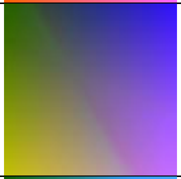
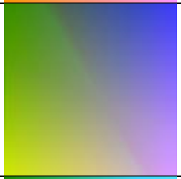
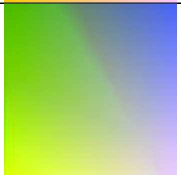
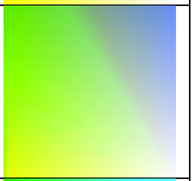
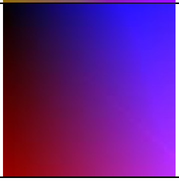
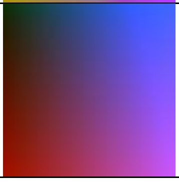
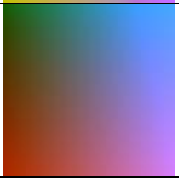
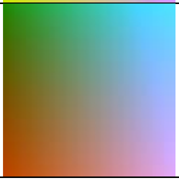
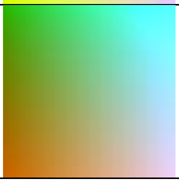
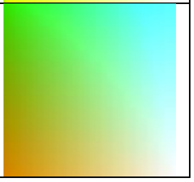
This is an unconstrained nonlinear least-squares problem that requires an iterative method to optimize it. The authors used the Mathematica function FindMinimum with the Levenberg-Marquardt algorithm to compute the values for each pixel. The starting values were chosen to be the L*, a*, b* values of the left eye. As in LS and UN, the unconstrained problem produces *r*, *g*, and *b* values that lie outside the RGB color solid. Clipping is used to ensure that all colors lie inside the RGB cube. We discuss this below.

3. RESULTS

3.1 Color comparisons when $C_{left} = C_{right}$

To gain intuition for how the colors changed based on the approximation, an initial study was done to examine the colors that were produced when the left and right eye colors were the same. We created images from cross sections of the RGB cube varying the green component from {0, 50, 100,…250} and blue and red from {0, 1, 2,…255}. The results were sufficient to conclude that the CIELab approach is superior to the other two approximation techniques in color faithfulness, particularly in the red area. Applying the same technique to blue or red cross sections produced similar conclusions. In Table 1 we show the results for the green cross sections. Note that they must be viewed through the glasses on an LCD display for comparison. The results for the Wimmer methods and LS were inferior to UN and CIELab and are not included. We will give examples of all methods in the appendix.

Table 1: Green cross sections of the RGB cube

G	0	50	100	150	200	250
Cross Section						
UN						
CIE-Lab						

Although this method preserves color "better" than the other approximation methods discussed, we noted above that hues are modified by all of the approximation methods.

Parallelizing the computation is recommended as is finding good starting values for the iteration. The optimal value for adjacent pixels should be considered since images usually have color concurrency. We have not tested this. Constraints can be added such as limiting color intensities, bounding saturation, etc. In our examples constraints were not considered.

3.2 Clipping

To better understand the behavior of the method and consider the clipping issue, we calculated vector fields that show how colors are mapped in the RGB space. In Figure 1 we present vector diagrams that illustrate the behavior for the G = 0 face. The inner cube is the RGB cube. The black corner is indicated by a dot.

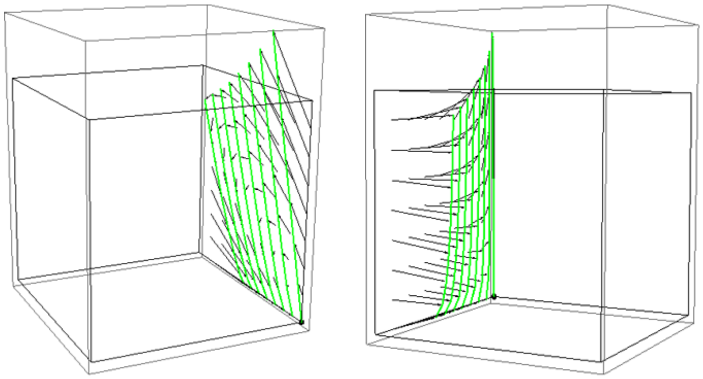
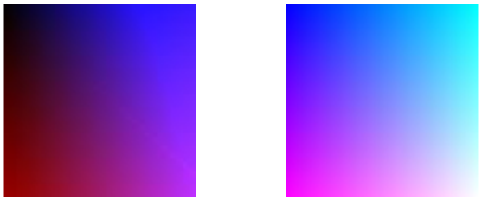


Figure 1: Vector maps of CIELab color modifications for colors on the G = 0 face.

In an unconstrained optimization some colors are mapped outside the RGB cube (this phenomenon also occurs in LS and UN). A simple and effective way to force all colors to lie in the RGB cube is to clip after optimization: all colors mapped outside the cube are modified to lie on the surface of the cube: if any coordinate is greater than 255 we set it to 255. If any coordinate is less than 0 we set it to 0. This can create "creases" or color discontinuities in the color slice. This can be observed in the CIELab color squares in Table 1.

For the case $g = 0$, the upper left corner of the color square in Table 1 is the (0, 0, 0) or black point in the RGB cube. As we move upwards on the cube face towards $b = 255$ or to the right in the color square in Table 1, we see that colors are mapped above the $b = 255$ plane. Clipping will set the blue component to 255. In Figure 2 comparison of the color square and the $b = 255$ plan illustrate this color discontinuity caused by clipping. The colors in the upper left corner of the $b = 255$ plane match the colors in the upper right corner of the CIELab map for $g = 0$.



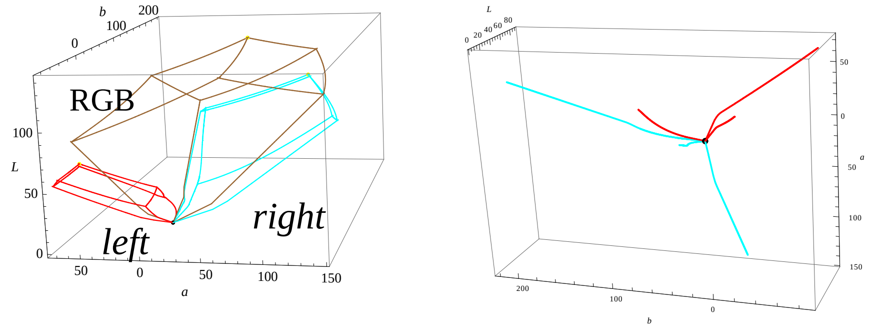
(a) (b)

Figure 2: Clipping example, (a) CIELab $g = 0$ plane, (b) $b = 255$ plane

Observe that clipping is not equivalent to "shaving" which maps a color to the first face penetrated by a vector. If one component is greater than 255 then the vector is shaved. If two components are greater than 255 then the color is mapped to an edge of the RGB cube. If all components are greater than 255 then the color is mapped to white. The order of clipping is not important. Equivalent assertions hold for components that are less than zero. Although adding bound constraints on r, g, and b in the approximation can preclude clipping, it increases computation time. Clipping is a faster alternative.

4. RETINAL RIVALRY

From the International Stereoscopic Union: “Retinal [binocular] rivalry is the simultaneous transmission of incompatible images from each eye” [03]. It was shown in [08] and in Figure 3 below that the gamut produced by the left and right eye filters for the red/cyan glasses and LCD displays used in this study are disjoint and hence retinal rivalry is guaranteed. It is not possible to match colors between the left and right eye views to totally eliminate binocular rivalry, contrary to the claim of some authors [16]. Indeed, in this case, the colors produced by viewing an LCD display through either eye filter cannot be reproduced on the display, so an examination by color matching of the color perceived by most people when viewing a color through the filters is problematic, particularly when taking into account the cycling that can occur in binocular rivalry [03].

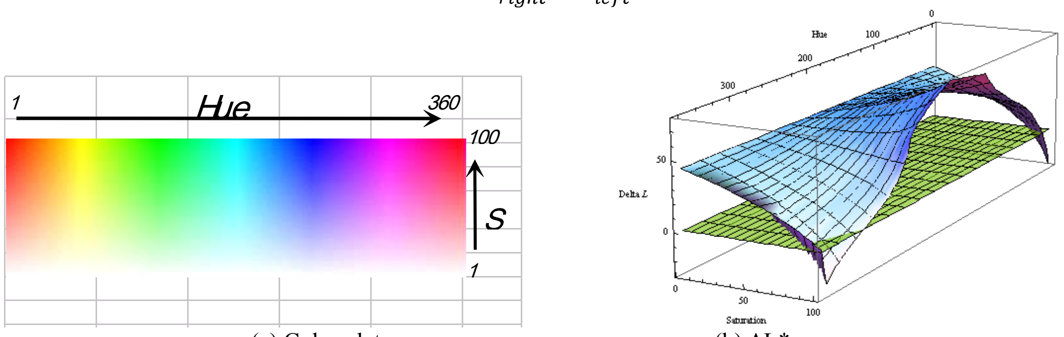


(a) (b)

Figure 3: (a) Color solid in CIEL*a*b*, for RGB, left and right eye filters, (b) Filter axes in CIEL*a*b* space

The worst form of retinal rivalry, however, occurs when the brightness difference perceived by the left vs. right eye is large, e.g., one eye sees a color close to black and the other sees a bright color, or one eye sees a color close to white and the other sees a bright color. For an example see Figure 5(a). The color is blocked by the red filter and the left eye sees black. In Figure 4(a) viewing the color plate by closing each eye sequentially illustrates this phenomenon. We will call this phenomenon *brightness disparity*. In CIEL*a*b* color space, L* is metric lightness, while a*, b* refer to the metric chromaticity. We propose a brightness disparity measurement using the following function:

$$\Delta L^* = L^*_{right} - L^*_{left}$$



(a) Color plate (b) ΔL*

Figure 4: Color plate (Hue=1~360˚, Saturation=1~100, Value=100)

We map the plate into the CIELab space for the left eye and right eye values. The value ΔL* reflects the brightness disparity perceived by the right eye vs. the left eye.

In the Appendix we give an example of a racing car (Figure 10 (a) through (f)) where considerable brightness disparity occurs for some of the hues. Anaglyph techniques that produce good color representation will often preserve the brightness disparity and indeed can make it worse. The goal is to modify all colors that produce a large brightness disparity. Whether to modify before or after computing the anaglyph has not been studied. Understanding how the

CIELab algorithm affects HSV values, described in 4.2 is also important. Below we mention two ways to approach this. We first describe a method proposed by Peter Wimmer for calculating anaglyphs that attempts to reduce rivalry.

4.1 Wimmer Maps

In [06] Peter Wimmer suggests the following maps to reduce rivalry, which he calls "Optimized Anaglyph." Wimmer's method discards the red component from the original images and replaces it with a red channel derived from the green and blue components.

$$\begin{bmatrix} r_a \\ g_a \\ b_a \end{bmatrix} = \begin{bmatrix} 0 & 0.7 & 0.3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} r_{left} \\ g_{left} \\ b_{left} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_{right} \\ g_{right} \\ b_{right} \end{bmatrix}$$

The Optimized Anaglyph reduced the retinal rivalry indeed, but obviously color reproduction is poor, especially for the red hue colors. In Wimmer's improved method [10] a gamma correction is applied to brighten up final red channel r_a that partially maps the red channels to green and blue before applying the above formula using the following:

$$g'_{left} = g_{left} + f_1 * \max(0, r_{left} - g_{left}) \qquad g'_{right} = g_{right} + f_1 * \max(0, r_{right} - g_{right})$$
$$b'_{left} = b_{left} + f_2 * \max(0, r_{left} - b_{left}) \qquad b'_{right} = b_{right} + f_2 * \max(0, r_{right} - b_{right})$$

$$\begin{bmatrix} r_a \\ g_a \\ b_a \end{bmatrix} = \begin{bmatrix} \left[\dfrac{1}{255}\left(191 * g'_{left} + 64 * b'_{left}\right)\right]^{\frac{1}{\gamma}} \\ g'_{right} \\ b'_{right} \end{bmatrix}$$

The values f_1 and f_2 are red mapping parameters while a gamma value γ is used to brighten the red channel. The values of these parameters are not dependent on the transmission properties of the filters or the spectral properties of the display. They can be adjusted according to the image. We use the values $f_1 = .45$, $f_2 = .25$, and $\gamma = 1.6$ as suggested by Wimmer [11]. We give examples in the Appendix of the Wimmer method with gamma correction.

4.2 Desaturation

In a red/cyan environment, brightness disparity occurs when a color is blocked by one filter and is perceived as dark while the other filter transmits a brighter color. To adjust the perceived brightness disparity without modifying the color suggests a move to the Hue (*h*), Saturation (*s*), and Value (*v*) (HSV) color system. The conversion equations are as follows. The conversions assume *r*, *g*, and *b* are normalized between 0 and 1. The value of *h* is measured in degrees. The values for *s* and *v* are often given in percentages.

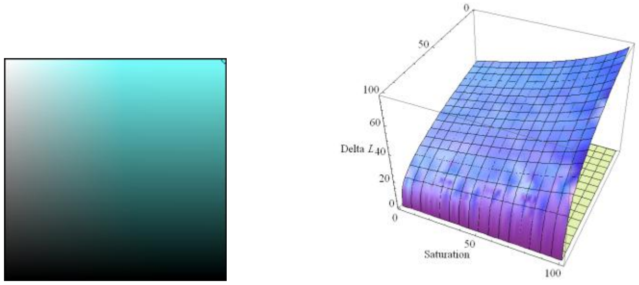
$$max = Max[r, g, b], \min = Min[r, g, b]$$
$$v = max, s = 0 \; if \; max = 0, else \; s = 1 - \left[\frac{little}{value}\right]$$
$$h = \left\{ \begin{array}{c} 0 \quad if \; max = min \\ \left[60\left(\dfrac{g-b}{max - min}\right) + 360\right] mod \; 360 \quad if \; \max = r \\ 60\left(\dfrac{b-r}{max - min}\right) + 120 \quad if \; max = g \\ 60\left(\dfrac{r-g}{max - min}\right) + 240 \quad if \; max = b \end{array} \right\}$$

The Inverse map becomes

$$hi = \left\lfloor \frac{h}{60} \right\rfloor mod6, \quad f = \frac{h}{60} - \left\lfloor \frac{h}{60} \right\rfloor$$
$$p = v(1-s), \quad q = v(1-fs), \quad t = v(1-(1-f)s)$$

$$(r,g,b) = \begin{cases} (v,t,p) & if\ hi = 0 \\ (q,v,p) & if\ hi = 1 \\ (p,v,t) & if\ hi = 2 \\ (p,q,v) & if\ hi = 3 \\ (t,p,v) & if\ hi = 4 \\ (v,p,q) & if\ hi = 5 \end{cases}$$

As an example, consider the HSV square in Figure 5(a) where full saturation is at the right edge and full desaturation (grayscale) is on the left edge. Brightness or *v* is measured vertically and decreases as we move from the top edge to the bottom edge (black). As one views the square by closing each eye while wearing the glasses, and as one moves away from the right edge horizontally, the hue *h* remains constant, the brightness *v* remains constant, and the saturation *s* is reduced; the color becomes *desaturated* and brightness disparity is reduced. Note that the function ΔL* given in Figure 5(b) is not linear.



(a) (b)

Figure 5: (a) Hue = 180, (b) ΔL* for Hue = 180

This behavior of ΔL* shown above does not hold for all hues. Desaturation can sometimes make ΔL* larger. As an example in Figure 6(a) we show the hue square for H = 288 and the plot of ΔL* is Figure 6(b). To study the behavior of ΔL* we plotted the surface for every hue from 0 to 360, 0≤ *s* ≤ 100 and 0 ≤ *v* ≤ 100. The behavior of the surfaces as we desaturate from *s* = 100 to *s* = 0 is as follows:

For hues between 0 and 81 the surface increases. That is, desaturation makes brightness disparity larger (L* for the right filter is larger than L* for the left filter).

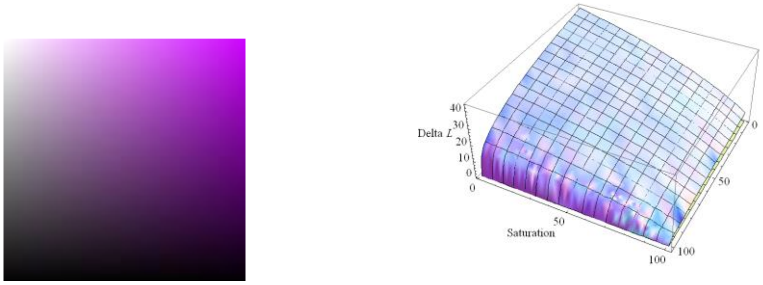
For hues between 0 and 18 the function ΔL* is negative for *s* close to 100. This also holds for hues between 302 and 360.

For hues between 82 and 230 the surface decreases. This implies desaturation decreases brightness disparity and reduces rivalry.

For hues between 231 and 245 the surface decreases and then increases. Hence, desaturation reduces brightness disparity for values of S close to 100. The point at which the surface begins to increase gets closer to 100 as the hue value increases.

For all hues from 246 to 360 the surface increases and desaturation makes the brightness disparity worse.

A reduction in *v* always reduces ΔL*. However, reducing *v* can affect the depth cue of shading in a three dimensional environment.



(a) (b)

Figure 6: (a) Hue = 288, (b) ΔL* for Hue = 288

It isn't clear whether desaturation should be applied to the left and right eye images before anaglyph computation or to the anaglyph. Application in the before stage requires a knowledge of how the hue will be affected by the optimization. On the other hand, the stereo may be affected if the anaglyph is desaturated.

Desaturation can also accentuate ghosting because of the increase in the intensities of complementary primaries. We seek an algorithm that will automate this process and give consistent results without destroying stereo or shading.

4.3 Gamma Correction

Viewing a subject through colored filters will reduce the observed color gamut of the subject as evident from Figure 3(a). As a result of color blocking by the filters, red may appear to be nearly black when viewed through the cyan filter (which transmits green and blue light). On the other hand, cyan hues may appear to be black when viewed through a red filter.

Inspired by Wimmer's improved method, in which a gamma correction is applied to brighten the red channel, we construct a modified algorithm based on gamma correction.

Gamma correction, also known as gamma nonlinearity, gamma encoding, or often simply gamma, is a nonlinear operation used to code and decode luminance or tristimulus values in video and in still image systems [17]. Normally, the same gamma correction value is applied to all RGB channels. In our algorithm, we choose different gamma values for each channel of RGB in the anaglyph image.

$$\begin{bmatrix} r'_a \\ g'_a \\ b'_a \end{bmatrix} = \begin{bmatrix} [r_a]^{\frac{1}{\gamma_r}} \\ [g_a]^{\frac{1}{\gamma_g}} \\ [b_a]^{\frac{1}{\gamma_b}} \end{bmatrix}$$

We choose a gamma value greater than 1 to brighten the red channel and a gamma value less than 1 to darken the green channel, so that the cyan color in an anaglyph image is brighter through the red filter and darker through the cyan filter. Gamma correction has little influence on brightness disparity for the blue channel and we assign a gamma value of 1 for this channel. In this paper, $\gamma_r = 1.6$, $\gamma_g = 0.8$, $\gamma_b = 1$ is used. The values were determined experimentally and are dependent on the properties of the glasses and the display.

We apply the gamma correction methods described above to the color plate in Figure 4(a). A plot of ΔL* after correction is similar to Figure 4(b). Hence, to show the changes in brightness disparity more effectively we plot $\left(\left|\Delta L^*_{before}\right| - \left|\Delta L^*_{after}\right|\right)$ in Figure 10. The results show that brightness disparity is reduced for most colors except some red hues. We note that the gamma correction used on computer displays by different PC manufacturers may also affect the results.

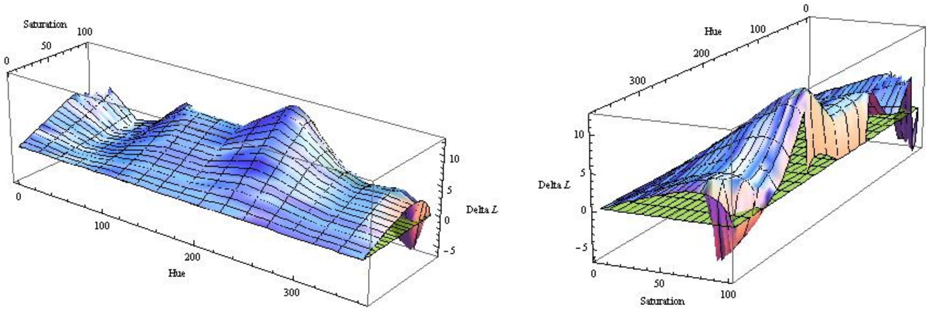


Figure 7: $\left(\left|\Delta L^*_{before}\right| - \left|\Delta L^*_{after}\right|\right)$

The value v of the color is preserved but the hue varies. Figure 8 shows the change in hue after gamma correction.

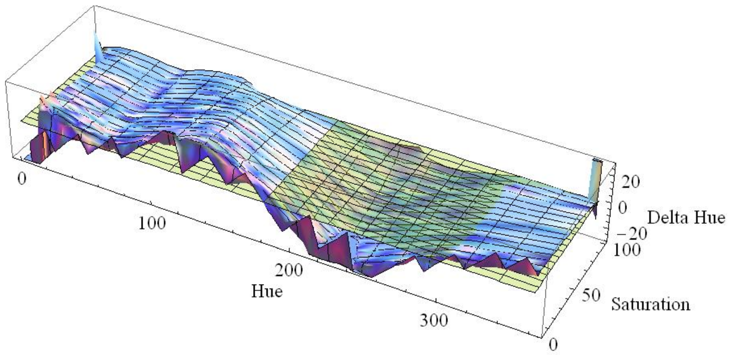


Figure 8: Difference of Hue surface before and after gamma correction: $Hue_{before} - Hue_{after}$

Unfortunately, gamma correction does not affect colors at the corners of the RGB cube with components 0 or 1. This suggests that applying desaturation when appropriate in concert with gamma correction can help to reduce rivalry.

5. SUMMARY AND CONCLUSIONS

We have presented and analyzed a method to compute anaglyphs based on approximation in the CIEL*a*b* color system. Colors are more faithful than those produced by the LS or UN methods. A negative aspect is the computational effort required. However, the method can be easily parallelized and starting values can be better selected. We seek a simple table lookup method by studying the inverse map (colors in the RGB cube that are mapped to an anaglyph color). These are areas for future research.

We presented a cursory examination of solutions to the retinal rivalry problem. Automating desaturation and gamma correction are subjects for future research.

We would like to thank Lee Filters of Burbank, CA for their help in providing transmission functions for anaglyph glasses.

REFERENCES

[01] Woods, S. J. and Tan, S. S. L., “Characterising Sources of Ghosting in Time-Sequential Stereoscopic Video Displays.” Proceedings Electronic Imaging 4660, 21-23 (2003).

[02] “retinal rivalry.” Merriam-Webster Medical Dictionary. 2009. Merriam-Webster Online. 12 Oct 2009 <http://www.merriam-webster.com/medical/retinalrivalry>.

[03] “retinal rivalry.” A Glossary of Stereoscopic Terms. 1997. International Stereoscopic Union. 15 Oct 2009 <http://www.stereoscopy.com/isu/glossary-r.html>.

[04] Zhang, Z. and McAllister, D. F., "A Uniform Metric for Anaglyph Calculation." Proceedings Electronic Imaging 6055, (2006).

[05] Dubois, E., "A Projection Method to Generate Anaglyph Stereo Images," Proceedings IEEE Int. Conf. Acoustics Speech Signal Processing , vol. 3, pp. 1661-1664, (2001).

[06] Wimmer, P. (2005). Anaglyph methods comparison. <http://3dtv.at/Knowhow/AnaglyphComparison_en.aspx>.

[07] "CIE Color Space." Federal Aviation Administration. 21 Feb. 2006 <http://www.hf.faa.gov/Webtraining/VisualDisplays/HumanVisSys2c4.htm>.

[08] Sanders, W. and McAllister, D. F., "Producing anaglyphs from synthetic images," Proceedings Electronic Imaging 5006, 348 – 358 (2003).

[09] Malacara, D., [Color Vision and Colorimetry: Theory and Applications], Bellingham: The International Society for Optical Engineering, 52 – 69 (2002).

[10] Wimmer, Peter. Personal Communication. “Anaglyphs”.

[11] Charles A. Poynton (2003). Digital Video and HDTV: Algorithms and Interfaces. Morgan Kaufmann. pp. 260, 630.

[12] Rychlik, W. “Cherries,” Pikes Peak Photo. < http://www.pikespeakphoto.com/cgi-bin/3Dviewer.cgi?pic=18>.

[13] Hannisian, R. “Pit,” Stereoscopic Imaging. < http://www.ray3d.com/pit_jpg.html>.

[14] McAllister, D. F. (Ed.) Stereo Computer Graphics and other True 3D Technologies, Princeton U. Press, Princeton, NJ, Oct. 1993

[15] McAllister, D.F. "3D Displays," Wiley Encyclopedia on Imaging, Jan., 2002, pp. 1327-1344

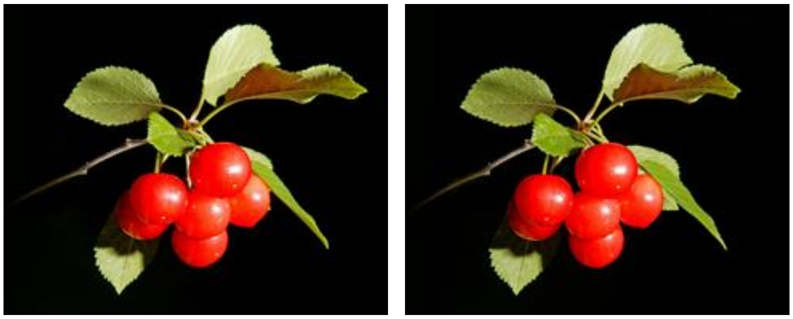
[16] Ramstad, M. J., “Display of generalized anaglyphs without retinal rivalry.” U.S. Patent 20,080,278,574, filed November 13, 2008.

[17] Artigas, J. M. and Felipe, A., “On the variation of the CIELAB coordinates with luminance,” Journal of Optics 18, 81 (1987).

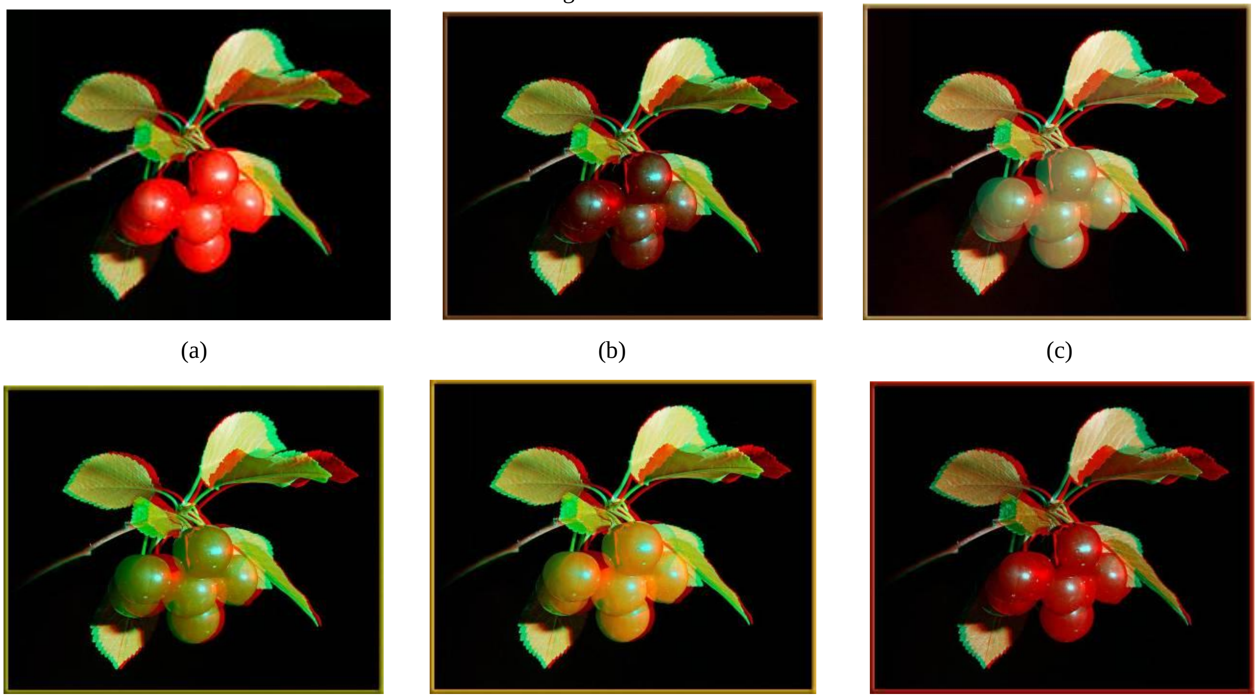
APPENDIX: EXAMPLES

Section I: Color fidelity

We observed that CIELab gives better representation of reds. Here we compare the PS (a), MPS (b), Wimmer (c), LS (d), UN (e), and CIELab (f) methods for the following image of a cherry. The stereo pair can be found at [12]. We note that while PS produces a bright red, the stereo is poor in that the cherries appear flat or without shape. We have not applied desaturation or gamma correction to any image in this section.



Original Stereo Pair



(a)

(b)

(c)

(d)

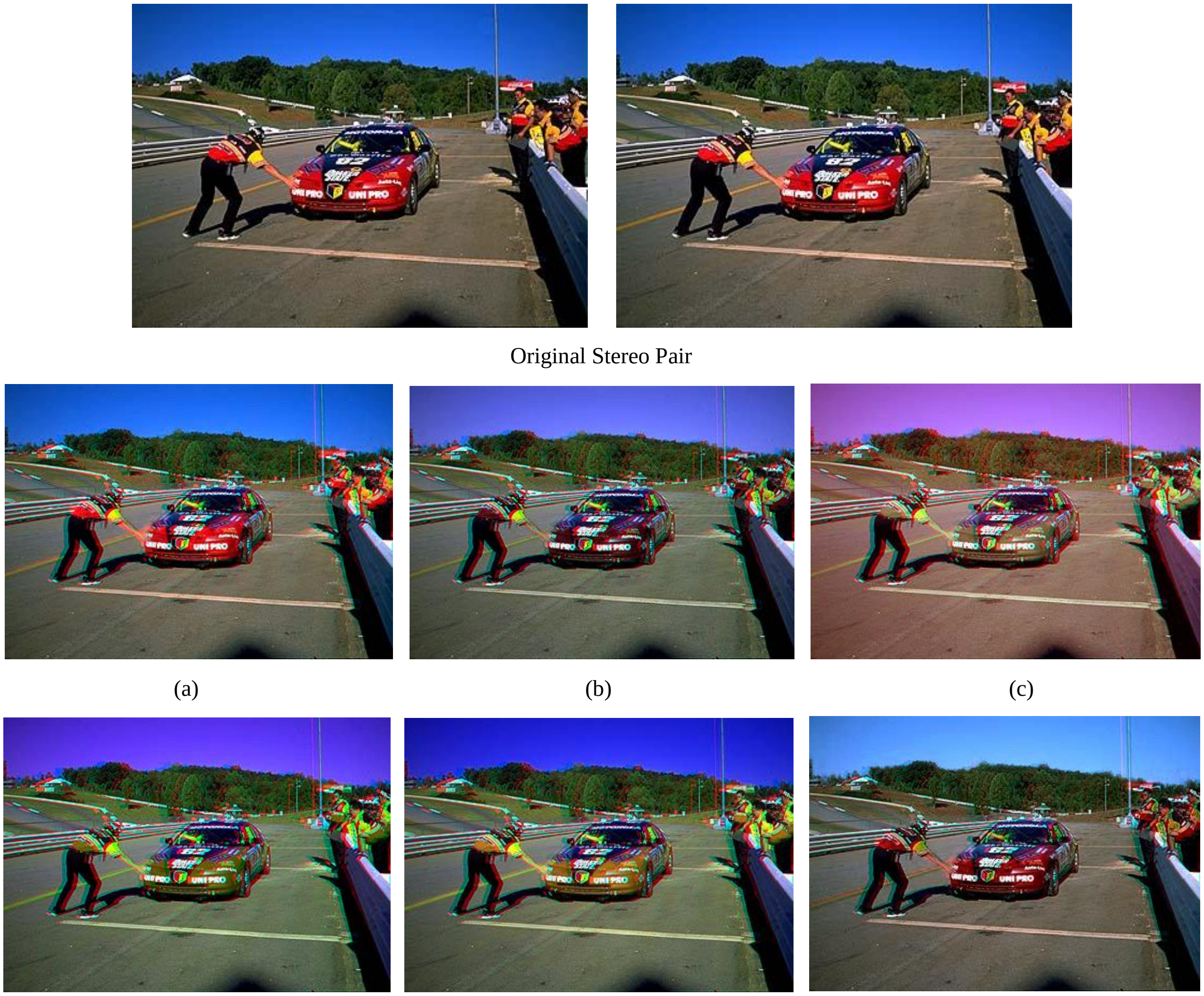
(e)

(f)

Figure 9

Section II: Rivalry

In the final figure we compare the PS (a), MPS (b), Wimmer (c), LS (d), UN (e), and CIELab (f) methods for the following image of a racing car. The image can be found at [13]. While wearing the glasses, viewing with each eye separately by closing the opposite eye shows clearly which colors produce rivalry in each method.



Original Stereo Pair

(a) (b) (c)

(d) (e) (f)

Figure 10