Color to Grayscale Conversion with Chrominance Contrast

Yuting Ye
University of Virginia

Figure 1: The sun in Monet's "Impression • Sunrise" has similar luminance as the sky. It can hardly be seen when the color image is converted to luminance values.

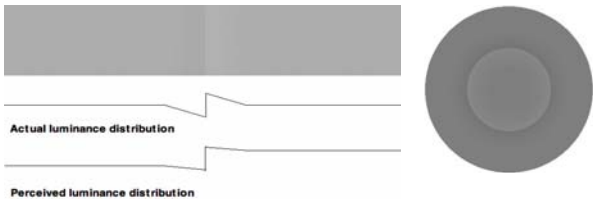

Figure 2: The Craik-O'Brien-Cornsweet illusions. The region adjacent to the light part of the "edge" appears lighter, and the region adjacent to the dark part of the edge appears darker

Abstract

Color information is usually discarded when converting a color image into a grayscale image. However, it is sometimes desirable to reflect chrominance change where luminance contrast is not obvious. Inspired by tone mapping technique, this paper proposes a similar method for color to grayscale conversion, which in addition takes into account the local chrominance changes. The result shows great improvement in efficiency over recently proposed global optimization techniques.

Keywords: Tone mapping, contrast enhancement.

1 Introduction

Color to grayscale conversion is in general a dimension reduction problem. Traditional method of retaining only luminance information does not work well when isoluminant colors present (Figure 1). Standard methods like Principle Component Analysis (PCA) and Local Linear Embedding (LLE) have been explored to tackle this problem, yet found not very effective ([Rasche et al. 2005a]). Rasche et al. [2005a, 2005b] and Gooch et al. [2005] have recently proposed optimization techniques to solve this problem independently. Their methods map isoluminant colors to different grayscale levels, aiming at minimizing the error in color distance between $L^*a^*b^*$ color space and grayscale, because the $L^*a^*b^*$ distance corresponds to perceptual distance of color. Both techniques look for a global solution to prevent artifacts between local regions. While having satisfactory results, their methods turn out not very efficient. They taek several minutes to convert a typical size color image, which is not acceptable for real world application.

However, we have some observations of this particular problem that can help develop faster algorithms. First, it is not well defined whether an isoluminant color will be perceived brighter than the other, because they are said to have the same brightness. Thus optimization techniques are questioned of their objective functions and further be questioned whether such expensive process is necessary. Second, images with clustered isoluminant colors usually span only a limited range, or a few small ranges. On the other hand, if they already cover the full range very well, there is little room to map isoluminat colors while preserving the correct order of luminance. Third, perception research shows that human eyes are easy to fool with illusions. Change between borders can cause illusion of false contrast (figure 2). As a result, we can argue that contrast enhancement with cues of chrominance change between borders is sufficient for this purpose. Our method thus uses tone mapping technique for contrast enhancement, and applies its local dodging-and-burning principle for chrominance changes.

Tone mapping ([Reinhard et al. 2002]) is used for tone reproduction of high dynamic range data in low dynamic range display devices such as

Figure 3 Some tone mapping results. High dynamic range data are obtained from Paul Debevec's radiance maps (http://www.debevec.org/Research/HDR/)

monitors or printers. It preserves fine details very well and is shown to perform better than many existing techniques. We will first briefly introduce the global and local operations of tone mapping, and then describe our modification to address for chrominance change, followed by discussions of some key issues and comparisons to other methods.

2 Tone Mapping

Tone mapping is a two-pass process. It first scales the luminance by the ratio of the key value to the log average luminance, and then scales the result again to displayable range (0, 1). This global operation can preserve details in low contrast areas, but not in high contrast area such as the very bright regions (figure 4). A dodging-and-burning operation is then applied for local tone reproduction. Figure 3 shows some tone mapping results with local operation.

2.1 Global Operation

The log average luminance is given by:

$$\overline{L}_w = \exp\left(\frac{1}{N}\sum_{x,y}\log\left(\delta + L_w(x,y)\right)\right) \qquad (1)$$

where $L_w(x,y)$ is the world luminance of the pixel in location (x,y); N is the total number of pixels in the image, and δ is a small value to prevent log of zero.

$L_w(x,y)$ is then scaled by :

$$L(x,y) = \frac{a}{\overline{L}_w} L_w(x,y) \qquad (2)$$

where α is the key value (i.e. middle tone) of the whole image. This key value controls the overall brightness of the image, and is mapped by the log average luminance. Its default value is set to 0.18 – the same as automatic exposure control in cameras. The result of difference choices of α can be found in Reinhard's tone mapping paper.

Next, we use (3) or (4) to bring $L(x,y)$ into displayable range (0,1):

$$L_d(x,y) = \frac{L(x,y)}{1 + L(x,y)} \qquad (3)$$

$$L_d(x,y) = \frac{L(x,y)\left(1 + \frac{L(x,y)}{L_{\mathrm{white}}^2}\right)}{1 + L(x,y)} \qquad (4)$$

L_{white} is the smallest white value, which will allow burn out effect if set to be lower than the maximum luminance. The use of L_{white} blends the scaling of $L(x,y)$ from 1 to $1/L$ for low and high luminance values respectively. It results in subtle contrast enhancement effect in high dynamic range images, and is of crucial importance when we later work with normal images in section 3.

Figure 4 Global operation alone cannot preserve fine details in very bright area (compared to figure 3).

2.2 Dodging-and-burning

The idea of dodging-and-burning is equivalent to specifying a different key value (α in equation 2) for every pixel according to its local neighborhood. The purpose is to bring up dark pixels and bring down bright pixels in high contrast regions. The scale of this local neighborhood can be different for each pixel. And the selection of this scale is the most important part in the algorithm. A center-surround function which mimics the difference of Gaussian filter is used to find the largest local region with little change of luminance.

First, a series of Gaussian kernels with different spatial scales are used to convolve with the luminance image before it is brought to displayable range:

$$R_i(x,y,s) = \frac{1}{\pi(\alpha_i s)^2} \exp\left(-\frac{x^2+y^2}{(\alpha_i s)^2}\right). \quad (5)$$

$$V_i(x,y,s) = L(x,y) \otimes R_i(x,y,s). \quad (6)$$

Where s is the scaling parameter controls the area of local region. Each scale level is 1.6 times larger than the previous one, starts from 1 pixel.

The center-surround function is then defined as:

$$V(x,y,s) = \frac{V_1(x,y,s) - V_2(x,y,s)}{2^\phi a/s^2 + V_1(x,y,s)} \quad (7)$$

where α is the key value we used in equation 2, and Φ is a color sharpening parameter. The difference is normalized by V_1 so it is independent of the absolute luminance, while the term $2^{\Phi}\ \alpha\ /s^2$ prevents numerical overflow when V_1 is too small. Later, we threshold V by ε, which is 0.05 by default, to define the local region scale s_m:

$$|V(x,y,s_m)| > \varepsilon \quad (8)$$

Finally, substitute L with V_1 in the denominator of (3) or (4), we get the displayable luminance:

$$L_d(x,y) = \frac{L(x,y)}{1 + V_1(x,y,s_m(x,y))} \quad (9)$$

$$L_d(x,y) = \frac{L(x,y) + \dfrac{L^2(x,y)}{L^2_{white}}}{1 + V_1(x,y,s_m(x,y))} \quad (10)$$

If we consider V_1 as the weighted average of a local region, it doesn't differ much from L in low contrast area, but makes a lot of difference in high contrast regions, scaling L differently as a result.

3 Chrominance Adaptation

Adopting the idea of dodging-and-burning for chrominance, we can modify equation (10) to enhance contrast between colors. First we need to define a chrominance value, which is in itself a dimension reduction problem because color is usually specified by 2 parameters. Since we use the luminance of CIE 1931 standard in previous calculation, we adopt its chromaticity diagram (figure 5) to define chrominance value as well. Other color space model or even user defined color mapping schema should work the same.

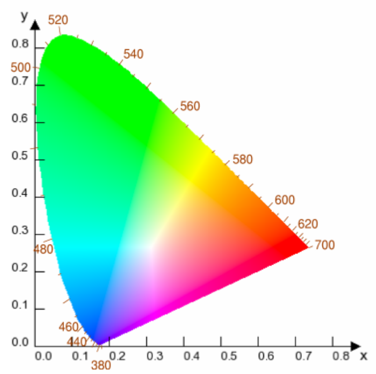

Figure 5 The CIE 1931 color space chromaticity diagram with wavelengths in nanometers. The reference white point is (x,y) = (1/3, 1/3)

We define chrominance as:

$$C = sqrt((x-1/3)^2 + (y-1/3)^2) \quad (11)$$

Note x, y here are the chromaticity coordinates, not the image coordinates as in other equations. This equation means further distance to the reference white, more colors (larger saturation). We compare the final result of this Yxy model with the $L^*a^*b^*$ model, and find no significant difference. Yet a more comprehensive chrominance model should work better.

Now we can convolve this chrominance with the same filter defined by (5):

$$V_c(x, y, s) = C(x, y) \otimes R_1(x, y, s) \qquad (12)$$

Equation (10) is then modified to reflect this factor:

$$L_d(x, y) = \frac{L(x, y) + \frac{L^2(x, y)}{L^2_{white}} + C(x, y)}{V_1(x, y, s_m(x, y)) + 1 + Vc(x, y, s_m(x, y))} \qquad (13)$$

As mentioned before, L_{white} has crucial importance to make the algorithm work. Since we are already working with luminance within the displayable range, the constant 1 in the denominator weights a lot compare to other terms. L_{white} together with the key value α determine the range and brightness of the final result. L_{white} is set to the maximum luminance by default, but they can be tuned to get better results for specific images.

With the same principle as dodging-and-burning, if large chrominance change presents in a local region, the difference in the nominator and the denominator will perturb the luminance to enhance contrast.

4 Result and Discussion

We implemented this algorithm in Matlab and obtain luminance as L= 0.265R+0.670G+0.064B. Four scales are used to calculate the convolution. While more scales make the result smoother, it also takes more time and has greater edge effect. Another thing to note is that the absolute values of luminance and chrominance may not be in the same order of magnitude. This problem becomes more obvious when using other color models to calculate chrominance. We need to scale it by the ratio of the two ranges. The effects of other parameters in tone mapping, as well as its comparison to other contrast enhancement methods are discussed in Reinhard's paper.

The prevailing advantage of our algorithm is in its efficiency, which is the same as tone mapping. Goodnight et al. ([2003]) has shown that this can be done in an interactive rate using GPU. Convolution is the most expensive operation in this algorithm. If we reduce the kernel size (the whole image here) and number of scales, performance can be further improved. In contrast, the optimization technique Gooch et al. proposed is $O(N^4)$ for a NxN image. They reported 204.0 seconds for a 200x200 image using an Althon 64 3200+ processor, and 25.7 seconds for the same image with implementation in GPU. Rasche et al. proposed a per-color based optimization technique. For an image with about 300,000 colors, it takes over 300 seconds. This is about a resolution of 480x640 because it is likely for each pixel to have different colors. Both of their techniques are far from interactive.

Successful and failure examples are shown in figure 6 and figure 7 respectively. There is no general technique that works for all the images. The way chrominance is calculated usually favors certain pairs of color but not the other. Because the perceptual model of color and brightness is not well reflected in existing color models we can only define distance between colors, but no ordering of brightness imply. Comparing the results of our algorithm and those of tone mapping only, we can see that it does not always enhance contrast as desired. It is because the sum of luminance and chrominance will sometimes cancel out each other, in a not desirable way. The fact that chrominance value is added to both nominator and denominator, instead of functioning as a scalar also causes difference in the final result. In general, our method can enhance contrast in the grayscale image and improve understanding of the color content.

We can always find a luminance model that maps colors to slightly different values, and than enhance contrast. Exact isoluminant colors are rare in real world images. Rasche's earlier method attempts to explore this fact to find an optimal linear transformation. This method will sometimes map shadows to bright white because no luminance ordering is enforced as constraint. When they later add this constraint to their optimization problem, time complexity kills all.

However, our argument is, when luminance difference exists, contrast enhancement will always make it look better.

One important effect of our algorithm is the nature of Gaussian filter creates smooth gradients around the border between two colors. This can be seen from a closer look of the images and from the histogram. This fits right in the illusion theory that contrast in the edge will imply overall difference in luminanc. We do not know it well enough to recreate this illusion, but it is obvious that outline of the border provides visual cue to differentiate the two regions in our result. We believe better knowledge of perceptual model can help further improve the algorithm.

References

GOOCH, A., OLSEN, S., TUMBLIN, J., AND GOOCH, B. 2005. Color2Gray: salience-preserving color removal. In *Proc. of SIGGRAPH'05*

GOODNIGHT, N., WANG, R., WOOLLEY, C., AND HUMPHREYS, G. 2003. Interactive Time-Dependent Tone Mapping Using Programmable Graphics Hardware. In *Proc. of the 14th Eurographics Symposium on Rendering,* pp 1-13

[a] RASCHE, K., GEIST, R., AND WESTALL, J. 2005. Detail preserving reproduction of color images for monochromats and dichromats. In *Proc. of IEEE CG&A Special Issue: Smart Depiction for Visual Communication*

[b] RASCHE, K., GEIST, R., AND WESTALL, J. 2005. Recoloring images for gamuts of lower dimension. In *Proc. of Eurographics*, 24, 3

REINHARD, E., STARK, M., SHIRLEY P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph. 21(3)*

Figure 6 Results that successfully differentiate different colors with similar luminance. Reference result can be found in Gooch's paper.

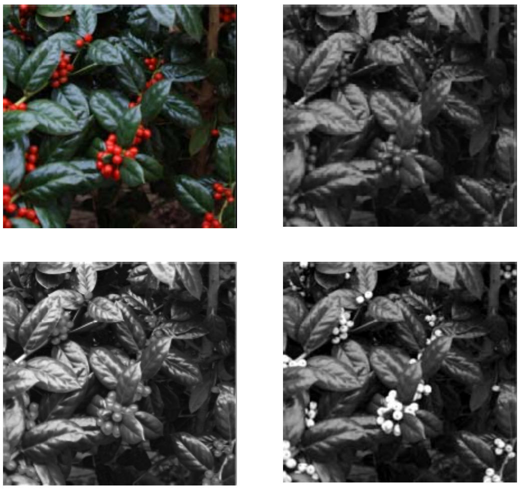

Figure 7 From left to right, up to down: color image, luminance image, our result and Rasche's result. Our result changes the luminance of red berries in a more subtle way than Rasche's result. Yet the improvement over the original image is discernible. Rasche's method is dedicated to help color deficient people, thus they change the contrast of red to green, blue to yellow in a more dramatic way.

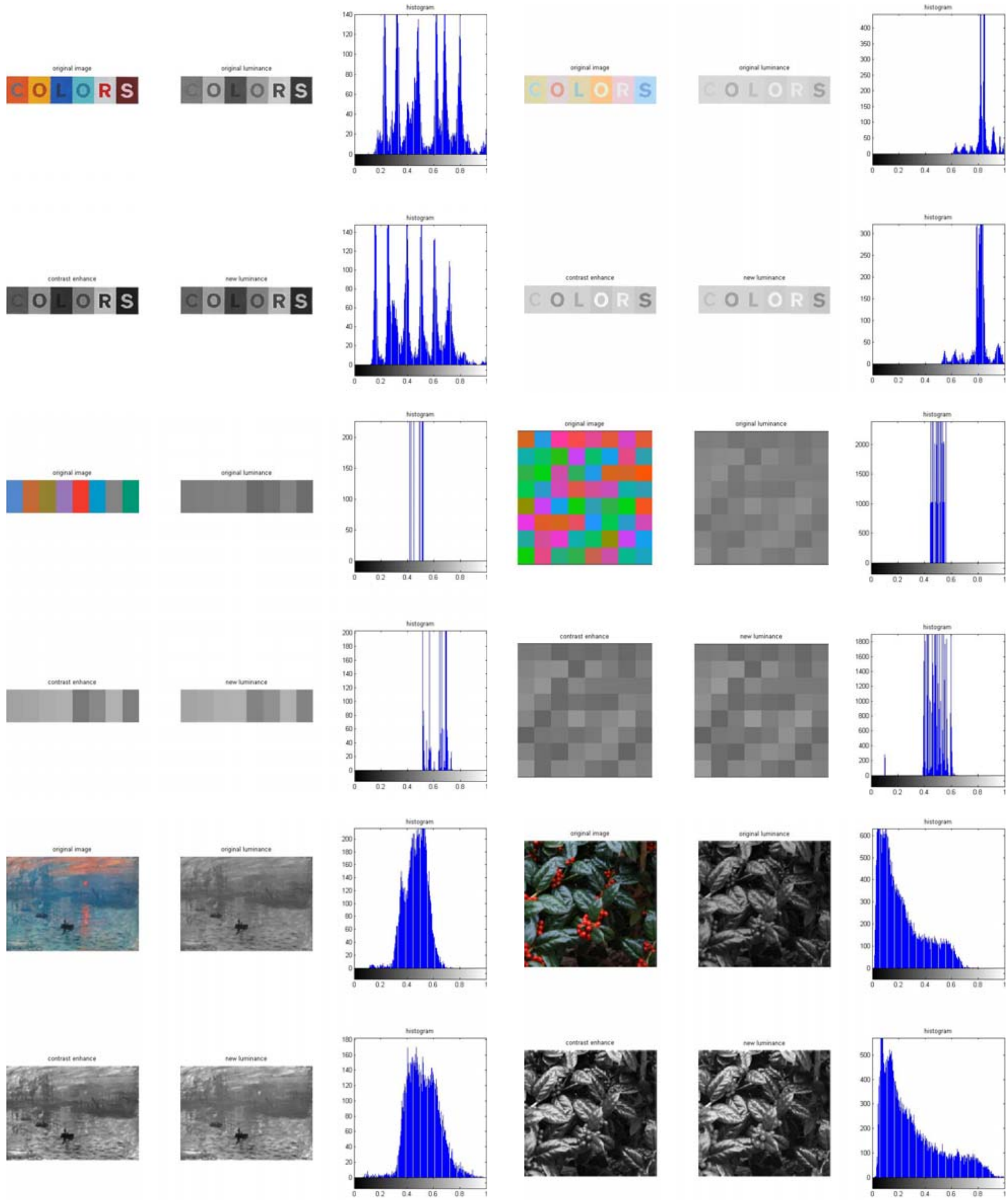

Figure 8 Each set of images from left to right, up to down, are the original color images, original luminance images, histogram of luminance, tone mapping results, our result and histogram of our result. We can tell clearly from the histogram that the range has been expanded while preserving the overall brightness of the image. Careful compassion with tone mapping results and our results can find the difference with respect to local chrominance change.