Geometrical Methods for Lightness Adjustment in YCC Color Spaces

Ramin Samadani, Guo Li[1]
Imaging Systems Laboratory
HP Laboratories Palo Alto
HPL-2005-191
October 20, 2005*

image processing, geometrical methods, image enhancement, color adjustment

Lightening or darkening an image is a fundamental adjustment used to improve aesthetics or correct exposure. This paper describes new geometrical algorithms for lightness adjustment, implementing fast traversal of colors along lightness-saturation curves, applicable when the data starts naturally in YCC space (JPEG images or MPEG videos). Here, YCC refers generically to color spaces with one luminance and two color difference channels, including linear YCC spaces and CIELAB. Our first solution uses a class of curves that allows closed-form computation. Further assuming that saturation is a separable function of luminance and curve parameter simplifies the computations. This approach reduces clipping and better adjusts lightness together with saturation. Preliminary evaluation with 96 images finds good subjective results, and clipping is reduced to about 5% of a prior approach.

* Internal Accession Date Only
[1]HP Imaging and Printing Group
Published in Color Image XI: Processing, Hardcopy, and Applications, 15-19 January 2006, San Jose, CA, USA
Approved for External Publication

Geometrical Methods for Lightness Adjustment in YCC Color Spaces

Ramin Samadani and Guo Li

HP Labs (Imaging Systems Lab) and HP Imaging and Printing Group
1501 Page Mill Rd., MS 1203, Palo Alto, CA 94304, ramin.samadani@hp.com

ABSTRACT

Lightening or darkening an image is a fundamental adjustment used to improve aesthetics or correct exposure. This paper describes new geometrical algorithms for lightness adjustment, implementing fast traversal of colors along lightness-saturation curves, applicable when the data starts naturally in YCC space (JPEG images or MPEG videos). Here, YCC refers generically to color spaces with one luminance and two color difference channels, including linear YCC spaces and CIELAB. Our first solution uses a class of curves that allows closed-form computation. Further assuming that saturation is a separable function of luminance and curve parameter simplifies the computations. This approach reduces clipping and better adjusts lightness together with saturation. Preliminary evaluation with 96 images finds good subjective results, and clipping is reduced to about 5% of a prior approach.

1. BACKGROUND

In RGB spaces, a simple and fast prior method applies a lightness function to each color channel. This method does not preserve relationships between colors. Two input colors with the same lightness, but one saturated and the other unsaturated, may result in different output lightnesses. There may even occur lightness reversals between two colors. In addition, the RGB color space is not natural for neither the commonly used compressed data formats such as JPEG for images and MPEG for videos, nor for devices such as printers and proposed luminance-chrominance projectors. [1]

In YCC, a prior method adjusts luminance using a linear or sigmoidal[2] function. This approach avoids the lightness reversals of the RGB method, but a significant disadvantage is the clipping artifacts when converting to other color spaces (RGB or CMYK). This approach also neglects the observation that the saturations of real objects vary when the light source intensity changes. Insufficient saturation may occur when dark colors are lightened and when light colors are darkened. Similarly, over-saturation may occur when dark colors are darkened and when light colors are lightened.

As seen in the color histogram of Figure 1, empirical observation of color histograms of individual objects in varying lighting shows that they lie on thin curves that connect the black point, a saturated color and the white point. Curves in different hue sectors may correspond to individual objects in an image, such as a person's skin tone, a blue shirt, or a green car. Complex factors, including specular reflection, imply that an object of a uniform material will result in colors along a luminance-saturation curve as the intensity and the angle of the light source changes. In the paper by Omer, [3] similar findings are used for computer vision applications. These findings motivate our approach, directly adjust colors by moving them along lightness-saturation curves while leaving hue unchanged. This approach keeps colors nearly in gamut, it avoids insufficient saturation when dark colors are lightened and when light colors are darkened, and it avoids oversaturation when dark colors are darkened and light colors are lightened.

Figure 2 shows two families of curves, controlling differently how saturation changes, for each of two hue planes. The vectors in the figure show the movement of two example colors, with the colors moving from the tails to the heads of the corresponding vectors. The adjustments work in color spaces, that we generically call YCC, consisting of a luminance and two color difference channels. The methods described here apply, for example, to both traditional, linear YCC spaces, to CIELAB and to other spaces.[4] This paper does not address trade-offs in color spaces, and applies the techniques to simple linear YCC spaces. Applying them in more uniform color spaces may offer additional advantages. Figure 3 shows a comparison of darkening using the new algorithm and the luminance only algorithm. The new method provides better results, viewed on a monitor, while substantially reducing clipping, as is shown by the small inserted images in the figure.

Geometrical methods are also used in the complex problem of gamut mapping, [5] where the emphasis is the projection of out-of-gamut colors onto gamut surfaces or within gamut volumes. In this paper, we address only lightening or darkening an image, where the improved clipping is also an important property of the methods developed.

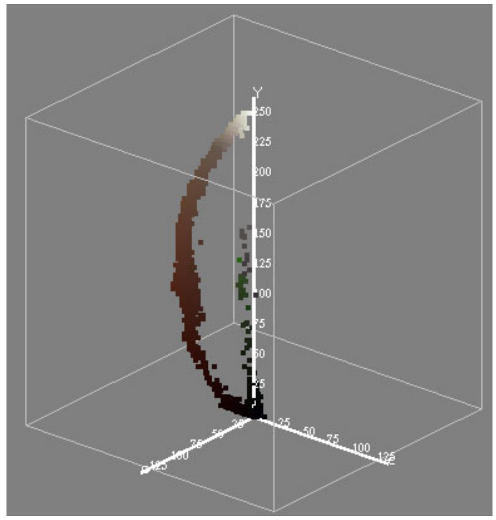

Figure 1. The color histogram for an image of a single object approximates a curve. The luminance is shown on the vertical axis and the color difference channels on the horizontal axes.

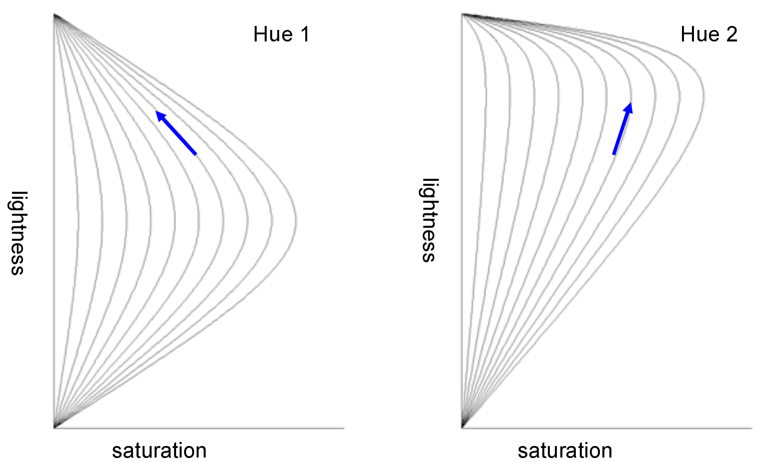

Figure 2. The proposed lightness adjustment moves each color along a curve in its saturation-lightness plane, with the hue unchanged. The corresponding curve families for two different hues are shown.

2. GEOMETRICAL DERIVATION

The notation in the following derivations include RGB colors, (r, g, b), with components in the range, $[0, 1]$. The YCC colors are given by (l, c_1, c_2). Here, l is the luminance, and c_1 and c_2 are the color difference values. The luminance l refers to either a luminance in a linear YCC space, or to a nonlinear luminance in the CIELAB space, for example. Finally, saturation s is given by $s = \sqrt{(c_1^2 + c_2^2)}$, and hue, h, is given by $h = tan^{-1}(\frac{c_2}{c_1})$.

For each hue, the easily met requirement for our algorithms is that $s = q(l, t)$, where s is saturation, and q is a function of luminance, l, and parameter, t. In general, the curves nearer or further from the gray axis, given by $s = 0$, may be of different shapes. One may imagine that the union of the curves for all the hues, and a particular parameter, t, form oddly shaped *onion skin* surfaces. Each pixel is conceptually processed as follows: 1) Calculate the hue; 2) Find the curve (parameter $t = \tau$) that the input pixel lies on; 3) adjust its lightness, l_i, using a function f, so that the output lightness l_o is

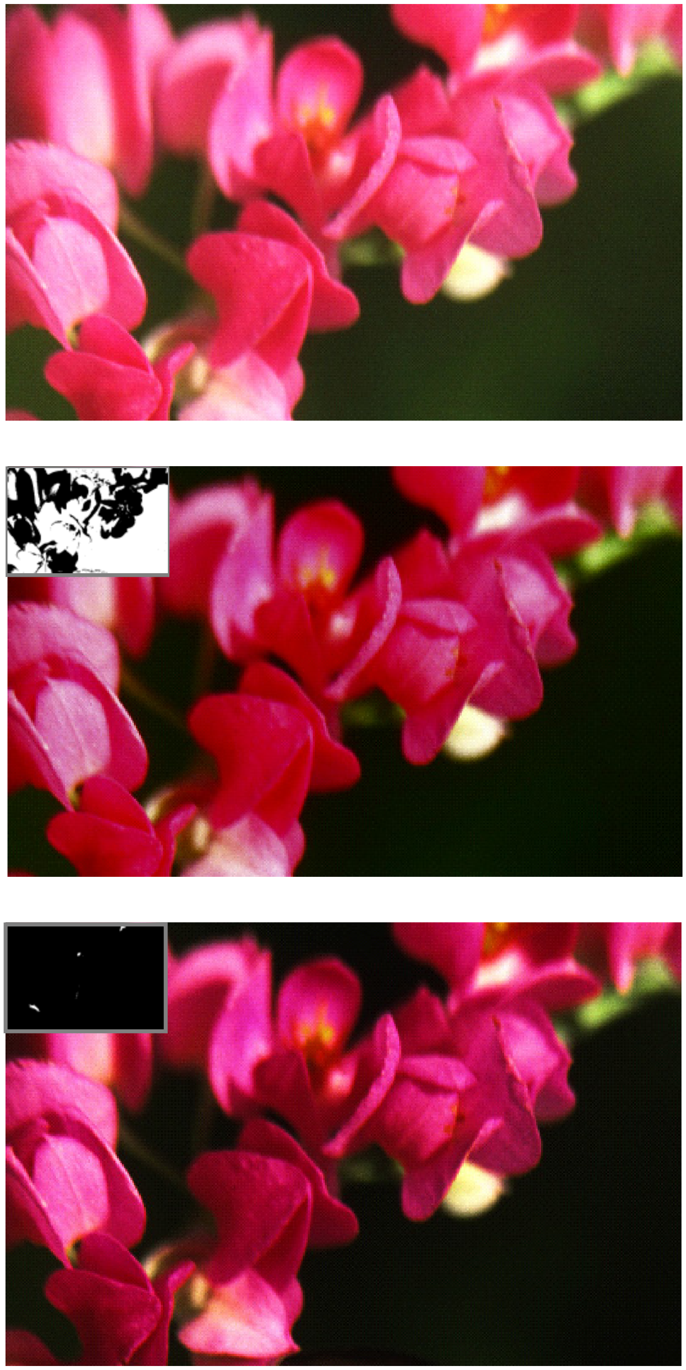

Figure 3. The original image is shown on the top. The traditional (Y only) darkening is shown in the middle (notice the flowers are over-saturated and loose spatial details). The new darkening is shown on the bottom. The small inserts in the darkened images show, in white, the pixels that clipped.

given by $l_o = f(l_i)$. For the experiments we used

$$l_o = f(l_i) = l_i + \beta sin(\pi l_i), \tag{1}$$

where β is a user-adjusted control for lightening or darkening; and 4) Find the output saturation s_o, by computing $s_o =$

$q(l_o, \tau)$.

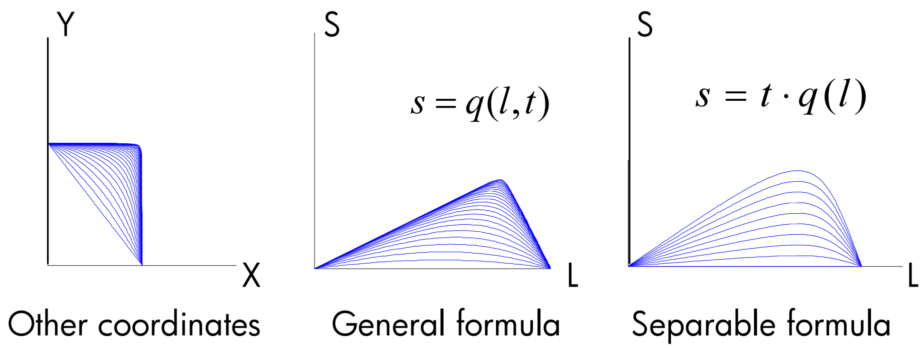

Figure 4. Curve families and coordinate systems used in the computations.

The design of the curve families has an impact on computational feasibility. This section describes the general design and Section 3 describes a simplified design. The curve families in this section have the interesting property that the lightness adjustment is made without leaving the gamut of RGB colors, but this holds only for linear YCC spaces, and it is not a crucial requirement of the method. Examples of such curves for a particular hue, are shown in the middle of Figure 4. It is important to note that the RGB color space is being used below only to assist in the derivation of the curve families for use in the YCC space. The output color conversion can be made directly to any other color space. Section 3 describes simplified curve families, such as the example shown on the right of Figure 4.

When the YCC color space and the RGB color space are related by a linear transformation, the RGB unit cube forms a parallelepiped in the YCC space, with diagonal along the l axis. Intersecting this parallelepiped with a half-plane of constant hue, h, determines a triangle with three (l, s) points: $(0, 0)$, $(l_a(h), s_a(h))$, and $(1, 0)$. In the example in the middle panel of Figure 4, the peak of the envelope of the curves corresponds, with implicit hue dependence, to (l_a, s_a).

The left panel of Figure 4 shows a curve family, each curve with a different parameter p, using the implicit functions $x^p + y^p = 1$ (Here, x and y are *spatial coordinates* unrelated to chromaticities or luminance. In this paper, l represents luminance). The triangle with vertex (l_a, s_a) determines a 2D affine transformation (which may be represented by a 3D homogeneous linear transformation)[6] from the (x, y) space shown in the left panel of Figure 4, to the (l, s) space, resulting in the curves plotted in the middle panel.

Computation with these curves is not possible in closed form. Proceed instead to approximate $y = (1 - x^p)^{\frac{1}{p}}$, with the rational functions,

$$y = \frac{a + bx}{c + dx}. \tag{2}$$

By satisfying the boundary conditions in the x, y coordinates, $y(0) = 1$ and $y(1) = 0$, the rational function form of Equation 2 simplifies to

$$y = \frac{1 - x}{1 - kx}. \tag{3}$$

This curve family, for k in $[0, 1)$, also allows lightness adjustment entirely within the RGB gamut, while allowing explicit solutions for the curve traversal computation.

The final derived formula is long, and it is not presented, but it is simply the solution of a quadratic formula. Here we present only the steps towards the solution. It can be shown that, given (l_a, s_a), the affine transformation from the (x, y) coordinates to the (l, s) coordinates may be written using homogeneous coordinates, [6]

$$\begin{pmatrix} x_o \\ y_o \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \frac{(1-l_a)}{s_a} & 0 \\ -1 & \frac{l_a}{s_a} & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} l_o \\ s_o \\ 1 \end{pmatrix}. \tag{4}$$

To find (la, sa), start with the transformation M, between YCC colors and RGB, such as the one we used,

$$\begin{pmatrix} l \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0.3008 & 0.5859 & 0.1133 \\ 0.5117 & -0.4297 & -0.0820 \\ -0.1719 & -0.3398 & 0.5117 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}. \tag{5}$$

With the knowledge that this transformation converts the RGB cube to a parallelepiped, the formula for (l_a, s_a) for any hue may be derived in terms of the (l, c_1, c_2) coordinates of the two primary or secondary RGB colors adjacent in hue. Then, equation 4 may be solved in closed form, for s_o, in terms of l_o, k, l_a and s_a, resulting in the solution of the quadratic.

This lightness adjustment was implemented and evaluated with several images, yielding good results.The hue dependence of (l_a, s_a) provides adequate control over change in saturation, even for colors like yellow that quickly change saturation with lightness. The simplification of the next section, however, yields similar results with simpler computations.

3. THE SEPARABLE ASSUMPTION

For each hue, instead of the general formulation, $s = q(l, t)$, now assume that saturation is a separable function of luminance l and parameter t, given by

$$s = tq(l). \tag{6}$$

This means $q(l)$ determines the shape of all the curves in the curve family, and the parameter t may be interpreted as a scale parameter. The right panel of Figure 4 shows examples of the separable curve family, for a number of different t values.

With an input color with saturation, s_i and luminance, l_i, Equation 6 results in $s_i = \tau q(l_i)$, where τ is the value of parameter t for the curve that passes through the input color. Since the output color is on the same curve, it is also true that $s_o = \tau q(l_o)$. Combining these facts, eliminating τ, and showing the results with explicit hue dependence,

$$s_o = \frac{q(l_o, h_i)}{q(l_i, h_i)} s_i = [\frac{q(f(l_i), h_i)}{q(l_i, h_i)}] s_i \equiv \alpha(l_i, h_i) s_i, \tag{7}$$

The lightness adjustment based on Equation 7, is shown with the simplified pseudocode in Figure 5. We found adequate control over the curve shapes by using Bezier curves[6] for q, with control points at (l, s) pairs $(0, 0)$, $(1, 0)$ and a double control point at (l_a, s_a). For a given hue, the parameters (l_a, s_a) are determined as before. Examples of Bezier curves for two hues were shown in Figure 2. For comparison, the standard algorithm is also shown in Figure 5.

The computations may be made efficient by precomputing α, the quantity in the square brackets of Equation 7. This precomputation is possible once a lightness function, f, and a user lightness setting β in Equation 1, are defined. Interestingly, if α is precomputed, it does not matter how complex the function q is, since it will only be evaluated during the precomputation. One may reduce the memory required by the lookup-tables by sampling hues for evaluation of $\alpha(l_i, h_i)$ and interpolating between the hues for a given input pixel hue.

4. RESULTS

Results are provided for the separable algorithm of Section 3. Informal evaluations were made using 96 RGB images of people, man-made structures and natural scenes. RGB images were used for ease of evaluation on a monitor. Each image was converted to YCC using Equation 5. Then both the standard and new algorithms shown in Figure 5, lightened the images using the lightness function of Equation 1 with lightness parameter $\beta = .25$. The images were converted back to RGB using the inverse of the matrix in Equation 5 and then clipped. The output images were then inspected on a monitor.

Subjectively, the majority of the outputs of the new method were preferred to the outputs of the standard method. Some of the improvements were due to reduced clipping of colors. Flesh tones, for example, often clipped using the standard method, but this clipping was substantially reduced with the new method. We also noticed that the darker colors in the images were lightened to more pleasing, saturated colors with the new method, than upon lightening with the standard method.

For some images, the output of the standard method was preferred. Examination showed this to be due to the accidental direction of the clipping of the standard method. For example, in a picture with a uniform blue sky, the new method

```
STANDARD ALGORITHM
for each pixel
    l_o = f(l_i) (adjust the lightness)
    c_1o = c_1i (do not modify the chroma)
    c_2o = c_2i (do not modify the chroma)
    convert l_o c_1o c_2o to output color space
    clip output color to be within range
end

NEW ALGORITHM
for each pixel
    compute h_i
    l_o = f(l_i) (adjust the lightness)
    α(l_i, h_i) = q(l_o, h_i)/q(l_i, h_i)
    c_1o = α(l_i, h_i) * c_1i
    c_2o = α(l_i, h_i) * c_2i
    convert l_o c_1o c_2o to output color space
    clip output color to be within range
end
```

Figure 5. The standard luminance-only algorithm and the new algorithm.

lightened the image and desaturated the sky color, whereas the standard method clipped this color to a bluer output color was preferred. However, in the standard method the clipping direction is uncontrolled, so that slight changes to the input image, for example, a sky image with changing sky colors, would have resulted in a less pleasing image.

We also generated clipping masks for each output, similar to the inserts in the example of Figure 3, with pixels that clipped shown in white. The proportion of clipped pixels of the new method versus the standard method was only $5.46\% \pm 2.07\%$, where the standard error of the proportion was estimated using the bootstrap [7] with 10000 repetitions. This confirms that the new method clips much less than the standard method.

REFERENCES

1. R. W. G. Hunt, "Displays: past, present and future," in *Proceedings of the 11th International Displays Workshop*, pp. 1–3, (Tokyo, Japan), December 2004.
2. G. Braun and M. Fairchild, "Image lightness rescaling using sigmoidal contrast enhancement functions," *Journal of Electronic Imaging* **8**, pp. 380–393, October 1999.
3. I. Omer and M. Werman, "Color lines: Image specific color representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 04)*, **II**, pp. 946–953, IEEE, June 2004.
4. A. N. Netravali and B. G. Haskell, *Digital Pictures*, Plenum, 2nd ed., 1995.
5. J. Morovic and M. R. Luo, "The fundamentals of gamut mapping: a survey," *Journal of Imaging Science and Technology* **45**(3), pp. 283–290, 2001.
6. I. Faux and J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood Limited, 1979.
7. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 2nd ed., 1992.