

oRGB: A Practical Opponent Color Space for Computer Graphics

Margarita Bratkova*

Solomon Boulos†

University of Utah

Peter Shirley‡



Figure 1: Visualization of the oRGB decomposition of the image on the left into luma, yellow-blue, and red-green channels.

Abstract

We present a new color model, oRGB, that is based on opponent color theory. Like HSV, it is designed specifically for computer graphics. However, it is also designed to work well for computational applications such as color transfer, where HSV falters. Despite being geared towards computation, oRGB's natural axes facilitate HSV-style color selection and manipulation. oRGB also allows for new applications such as a quantitative cool-to-warm metric, intuitive color manipulations and variations, and simple gamut mapping. This new color model strikes a balance between simplicity and the computational qualities of color spaces such as CIE L*a*b*.

1 Introduction

Digital images are stored, manipulated, and transmitted in a variety of color spaces. Many different color spaces have been proposed, with *RGB*, *CYMK*, *HSV*, and *CIE L*a*b** being especially popular (see [Hunt 2004; Fairchild 2005]). The *RGB* space is an additive color model and is probably the most commonly used among computer graphics practitioners, but it is not very intuitive. *CYMK* is a subtractive color model and is most useful in color printing. A bijective transform of *RGB* space, *HSV* was introduced by Smith [Smith 1978] as a more convenient and meaningful color space, designed specifically for computer graphics. At about the same time, the need for a device independent, perceptually linear color space led to the development of *CIE L*a*b**.

The recent interest in automatic image manipulation techniques such as color harmonization, color transfer, colorization, as well as the proliferation of work in computational photography, suggest a need for an intuitive, yet simple and practical computational computer graphics color space. While *HSV* has proven extremely useful in practice, there are two characteristics it lacks from our standpoint. First, its primary colors (red, green, and blue) are based on the mechanics of additive color mixing rather than the more natural psychological primary colors. Second, *HSV* is well-suited for color selection, but is not a good space for computational applications.

In this paper we introduce a new color space, the *oRGB* model. Like *HSV*, it is an invertible transform from *RGB*. The primaries

of this model are based on the three fundamental psychological opponent axes (white-black, red-green, and yellow-blue). The model has three channels - one luma (see Section 2.5 for a definition) and two chroma, as shown in Figure 1. The non-linearity of these channels relative to intensity uses the non-linear gamma encoding of the native *RGB* space. Thus it has some of the advantages of non-linear perceptual spaces such as *CIE L*a*b**, while maintaining a simple gamut and computationally simple transform to and from *RGB*. Because the chroma axes of *oRGB* go from warm to cool (red to green and yellow to blue), *oRGB* has a quantitative concept of color warmth as needed by some artistic applications. This new space is reasonably simple, has intuitive axes, and is well-suited for computational applications.

In the rest of this work, we first review opponent process color theory and some of the color models used in computer graphics (Section 2). We then develop the details of the *oRGB* model (Section 3), and show the application of this color space for color adjustment, NPR lighting computation, color transfer, and gamut mapping (Section 4).

2 Background

We now review the high-level differences between various color ordering systems, paying particular attention to opponent color systems. We also establish several terms and symbolic conventions as the study of color spans several disciplines and there are no universally consistent uses.

2.1 Hue Spacing Systems

Most color spaces have one light-dark axis and two chromatic axes. The boundary of the realizable colors in the chromatic dimension is often displayed as a circle or “wheel”. The ordering of hues is the same in most color spaces, but the spacing between the hues differs. We summarize these to clarify the relationship between opponent color systems and more traditional hue ordering systems.

Four of the most popular types of color circles are shown in Figure 2. The two circles to the left are based on three primaries and how they mix. The uniform color circle, of which the one that visualizes the Munsell Color Space is perhaps the best known, attempts to arrange the hues so they are “evenly spaced”. Such uniform spaces often lack a concept of “primaries” as uniformity precludes even spacing between named colors such as red and blue.

*bratkova@cs.utah.edu

†boulos@cs.utah.edu

‡shirley@cs.utah.edu

To the right is a Hering-style opponent circle with four primaries on two natural axes. It emphasizes his concept of red-green and yellow-blue axes, while maintaining the order of colors as in the other color circles.

2.2 Hering's Opponent Process Color Theory

The opponent process theory of color was first advocated by Ewald Hering in the 1870s. It took a sharp departure from the prevalent theory of the time, the *trichromatic theory* of Young-Helmholtz, by proposing four hue primaries: red, green, yellow, and blue, instead of the traditionally accepted three: red, green, and blue.

While three primaries are needed to produce all possible colors, Hering noted that our subjective experience of color suggests that perceptually there is an encoding for the fourth. For example, while purple seems to be a mixture of red and blue, yellow does not seem to be composed of any other primaries. Instead, it seems to be a “pure” color, a primary itself. In addition, Hering observed that we don’t seem to have a phenomenological experience for a color that is both yellow and blue, nor red and green. However, we have one for a color that is both yellow and red (orange).

These observations led Hering to hypothesize that human color perception uses two chromatic axes: one that ranges from red to green, and another from yellow to blue. If these axes are independent, then people will be able to see a red and yellow color simultaneously (orange), but not a red and green color, nor a blue and yellow color, as they share the same “channel” (see Figure 3).

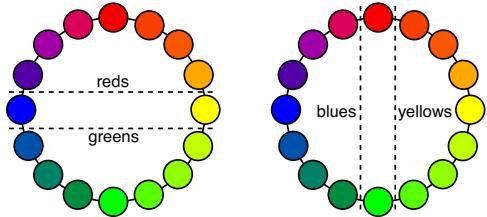


Figure 3: The color circle divided into its opponent sides. The top part of the circle is composed of colors that have a red tone in them, the bottom - green, the right - yellow, and the left - blue.

Hering’s ideas were controversial at the time. However, there has been a wealth of psychological and physiological evidence since then (Hurvich [Hurvich and Jameson 1957] brought the revival of the theory) that suggests that the organization of human color perception is at least partially based on opponent axes [Conway 2002; Valberg 2005]. Color scientists have moved more and more towards including some form opponency into many of their color spaces, with *CIE L*a*b** and the Natural Color System (*NCS*) being notable examples.

2.3 Opponent Space Models

A variety of spaces are based on opponent axes, with the television encoding *YIQ* being perhaps the best known. However, we make the distinction that most of these spaces are not Hering-style opponent spaces in that their principal axes do not include a red-green dimension (while some of them do include a blue-yellow). This is illustrated in Figure 4 for *YCC* (a modern digital descendant of *YIQ*), *lαβ* (a color space based on natural image statistics), and *CIE L*a*b** (an approximately perceptually uniform space). Note that

all of these models have more of a magenta-cyan axis than a red-green one. An opponent space proposed for computer graphics by Naiman [Naiman 1985] is similar to *YCC* (thought it has a slightly different matrix, and operates on linear *RGB*), but it also has an axis shifted toward magenta-cyan.



Figure 4: Visualization of the chroma channels for (left to right): *YCC*, *lαβ*, and *CIE L*a*b* color spaces. Compared to our new *orGB* space (Figure 1), these spaces do not have a true red-green channel.

The most well-known opponent space with a true red-green axis is the *NCS*. Unfortunately, there are no accepted color formulas to allow this space to be used for computation [Kuehni 2004], and is therefore not well suited for computer graphics applications.

We are aware of only three quantitative Hering-style opponent models. The first is that of Schwarz et al. [Schwarz et al. 1987] who embed their space in *XYZ* tristimulus space with non-orthogonal but true opponent axes. The second, introduced by Ware and Cowen [Ware and Cowen 1990], places the opponent primaries at corners of the model. Unfortunately, the axes are not opponent. The third is that of Shirriff [Shirriff 1993] who developed an orthogonal red-green and yellow-blue space, using *HSV* as an intermediary space. Each of these spaces were designed for color selection, are defined in terms of linear *RGB*, and are not well suited for computational color applications. We will explore this further in Section 4.

2.4 Limitations of Color Models

One approach to generating a color model is to proceed directly from existing scientific knowledge. While one might expect color science to be both a mature and static field, this is far from the case. The details of the physiological origin of color perception remain enigmatic despite the increasing understanding of opponent mechanisms [Conway 2002; Valberg 2005]. The psychophysical approach that is used in the construction of most models also has limitations. There is unlikely to be a “right” choice, as different color spaces have different goals and are all simplifications of the real experience [Kuehni 2003]. For this reason, existing color spaces are based at least partially on heuristics and differ largely in their goals and how they manage trade-offs.

Opponent color theory presents many unknowns and asymmetries. For example, it is believed that the opponent signals are not simple linear combinations of cone signals as is often diagrammed in textbooks [Chichilsky and Wandell 1999; Valberg 2005]. In addition, the axes do not all combine in symmetrical ways. The combinations of some axes create “important” colors with their own names, while others do not [Hardin 1998]. For example, red and yellow make orange, while green and yellow make chartreuse (which most English speakers associate with a pink or purplish color). Fundamental colors such as blue still look “blue” when darkened, while some may change color category such as an orange turning to “brown” when darkened. In addition, the chromatic axes have a natural symmetry

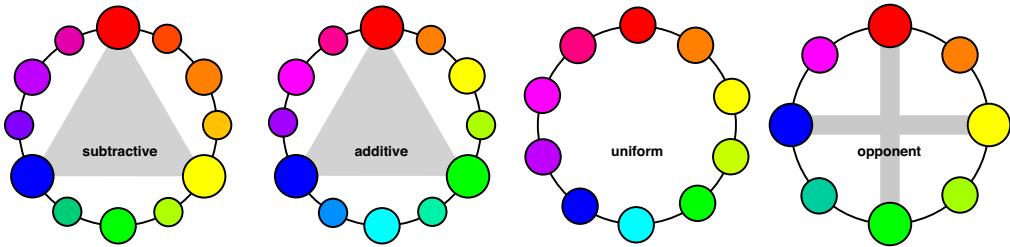


Figure 2: Visualization of four common types of color circles. The ordering of hues is preserved in each, while the position of the hues on the circle varies. The subtractive circle is that used by painters while a cyan-magenta-yellow circle (not shown) is used by most ink printing processes.

around grey while the achromatic black-white axis is believed to be (and represented in most color models as) just positive [Volbrecht and Kliegl 1998; Kuehni 2003].

These uncertainties and trade-offs suggest that there is no “true” color space. It is important that one bases the choice of color space on its performance in the applications of interest. Here we propose a color space that is suitable for computational applications and is simple, intuitive, and practical.

2.5 Terminology and Notation

Most computer graphics images are stored in some form of an *RGB* color space. While there are many different *RGB* spaces, we assume that for the most part, our images are encoded in *sRGB*, an increasingly popular standard. An often confusing notational issue is distinguishing between linear and non-linear intensity red, green, and blue components. A linear red channel stored with range $[0, 1]$ for example, would represent half the maximum physical intensity with 0.5. A non-linear one, however, might do so at 0.75, depending on the coefficients used.

We adopt the often used convention that a prime ($'$) is used to denote a quantity that is non-linear in intensity and ready for display. So the non-linear $R'G'B'$ stored in most files contains (R', G', B') , while a linear intensity RGB contains (R, G, B) . For *sRGB* the relationship between components is well approximated by $R' \approx R^{1/2.2}$. When converting to linear photometric quantities such as (X, Y, Z) , the linear (R, G, B) should be used.

Linear color spaces usually use *luminance*, a linear photometric standard for achromatic light response, and often denoted by the symbols L and Y . The value for luminance can be computed as a linear transform of a color value stored in linear color space. For example, for linearized *sRGB* values, the luminance is:

$$L = 0.213R + 0.715G + 0.072B \quad (1)$$

Computing luminance as a linear combination of (R', G', B') is not possible. Instead, a non-linear analog of luminance called *luma* is often used. There is no single standard for luma computation (see Poynton [Poynton 2004] for an excellent detailed discussion of different luma computation standards). Sometimes the symbols L or Y are also used for luma, but we use L' to distinguish it from luminance. Typically, luma is a linear combination of (R', G', B') . For example the YCC standard uses:

$$L' = 0.299R' + 0.587G' + 0.114B' \quad (2)$$

The non-luma axes typically encode chromatic information, and are also linear combinations of (R', G', B') . The color quantities are usually called *chroma*.

The *gamut* of an *RGB* monitor is the set of colors it can physically reproduce. In *RGB* space this gamut is the unit cube $[0, 1]^3$. When a color space has a well-defined mapping from *RGB*, the *RGB* gamut within that space is the shape created by sending the entire *RGB* cube through that mapping. For some color spaces the boundaries of that *RGB* gamut are algebraically simple, e.g. the *RGB* gamut of the *YIQ* space is a parallelepiped. For others such as *CIE L*a*b** the surfaces on the *RGB* gamut boundary are algebraically complicated. When color modifications take colors outside the *RGB* gamut, some method must be used to map these colors back into gamut before they can be stored in typical *RGB* files. Such a method is usually called *gamut mapping* (for an overview see [Morovic and Luo 2001]) and component by component truncation is often used [Stone et al. 1988].

3 The *oRGB* Color Space

As our interest is an opponent color space that is ideal for *RGB* computation, we begin with Alvy Ray Smith’s original observation in his classic *HSV* paper:

Full-blown color theory is a quite complex subject, involving physics, psychology, and physiology, but restriction to the *RGB* monitor gamut simplifies matters substantially [Smith 1978].

Smith went on to design *HSV* as a simple bijective transformation of *RGB* that is useful for color picking, as it has natural axes of lightness, hue, and saturation. We would like an analogously simple space that retains these intuitive notions, but also uses the natural axes of light-dark, red-green, and yellow-blue as suggested by perceptual theories. Unlike *HSV*, however, we would like to be able to perform well in computational applications.

Inspired by Poynton’s framework [Poynton 2004] for how different color models deal with CIE tristimulus values, we note how two popular color spaces, *CIE L*a*b** and *lαβ* operate. Both of these spaces have been used effectively in computational settings, and unlike other opponent-like color spaces, store quantities that are non-linear in intensity. However, both of these spaces have complicated pipelines as shown in Figure 5. Each of them first “linearizes” the (R', G', B') to (R, G, B) , does a linear transform to an intermediate linear intensity space (either *XYZ* or *LMS*), applies a component-by-component non-linear map (either logarithmic or polynomial), and then does a linear transform to get into an opponent-like space.

Our goal is to design a space that is useful for graphics applications. Having a non-linear luma channel is more perceptually uniform than luminance, and we believe it is simpler, improves computational speed, and also works better for practical applications (see

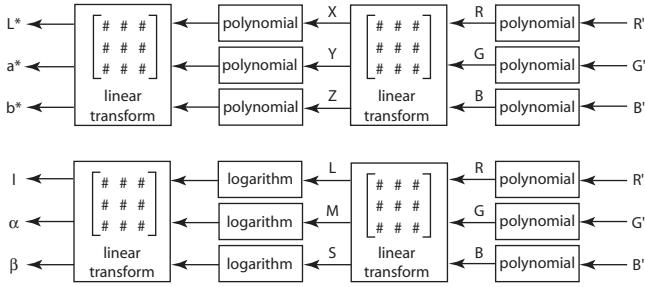


Figure 5: The pipelines of $R'G'B'$ to CIE $L^*a^*b^*$ (top) and $R'G'B'$ to $l\alpha\beta$ (bottom).

Figure 19 in Section 4.3). Therefore, instead of going through a device-independent space and reintroducing non-linearity, we simply apply a linear transform directly to the non-linear $R'G'B'$ values to get to an intermediate linear space we denote as $L'C'C'$. Like the CIE $L^*a^*b^*$ and $l\alpha\beta$ chroma axes, $L'C'C'$ does not have true Hering-style opponent axes. Therefore, we introduce a final rotation around the L' axis that achieves a true Hering-opponent model - oRGB (see Figure 6).

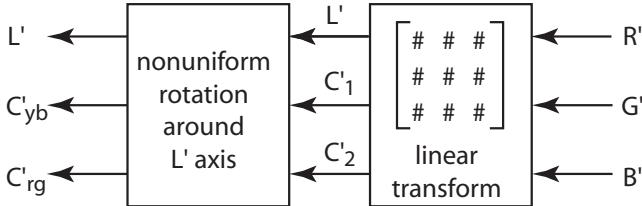


Figure 6: The pipeline of $R'G'B'$ (R', G', B') to oRGB (L', C'_{yb}, C'_{rg}).

3.1 oRGB Step 1: Linear transform of $R'G'B'$ to $L'C'C'$

There is an existing family of color spaces that are linear transforms of $R'G'B'$. These are those used for the video encoding of television systems including YIQ and YCC.

In a similar fashion, oRGB first transforms the RGB cube into a parallelepiped via a linear transform. This color space, $L'C'C'$, has 3 axes: white-black, yellow-blue, and magenta-red/cyan-green. The matrix for this transformation is:

$$\begin{bmatrix} L' \\ C'_1 \\ C'_2 \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5000 & 0.5000 & -1.0000 \\ 0.8660 & -0.8660 & 0.0000 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (3)$$

and its inverse:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.1140 & 0.7436 \\ 1.0000 & 0.1140 & -0.4111 \\ 1.0000 & -0.8860 & 0.1663 \end{bmatrix} \begin{bmatrix} L' \\ C'_1 \\ C'_2 \end{bmatrix} \quad (4)$$

3.2 oRGB Step 2: Non-uniform rotation around luma axis L' from $L'C'C'$ to oRGB

Is there an affine transform of the RGB cube that moves blue opposite yellow and red opposite green? Unfortunately, there is not. Figure 7 shows the RGB cube viewed along the white-black axis.

As can be seen there is no affine transform that will make red, white, and green collinear. The closest we can come is to put a red-magenta opposite a green-cyan.

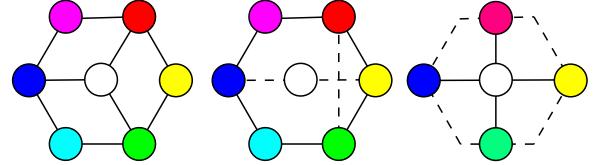


Figure 7: Orthographic view of RGB cube looking along white-black axis (left). No affine transform will put red-white-green in line (middle). The y-axis will have a red-magenta and a green-cyan as its extreme values (right).

To create a true opponent space where red is actually opposite green and orthogonal to the yellow-blue axis, we must apply some non-affine transform. We use the simplest one we can think of: a constant scaling of angle between the opponent directions (see Figure 8). A smoother mapping was attempted, but it made no visual difference, and so we chose the simplest method that worked well.

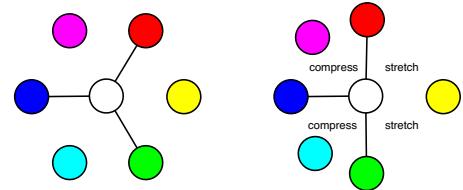


Figure 8: To take the red and green directions to the vertical axis, we compress angles on the blue side, and expand them on the yellow one.

The transformation from $L'C'C'$ (L', C'_1, C'_2) to oRGB (L', C'_{yb}, C'_{rg}) is just a compression or decompression of angles depending on which quadrant the linearly transformed point ends up in. For points above the yellow-blue axis, the angle in the linear chroma plane is $\theta = \text{atan2}(C'_2, C'_1)$. The new angle in oRGB space, θ_o is:

$$\theta_o(\theta) = \begin{cases} (3/2)\theta & \text{if } \theta < \pi/3 \\ \pi/2 + (3/4)(\theta - \pi/3) & \text{if } \pi \geq \theta \geq \pi/3 \end{cases} \quad (5)$$

To compute the point (C'_{yb}, C'_{rg}) in oRGB we simply rotate the (C'_1, C'_2) point:

$$\begin{bmatrix} C'_{yb} \\ C'_{rg} \end{bmatrix} = R(\theta_o - \theta) \begin{bmatrix} C'_1 \\ C'_2 \end{bmatrix} \quad (6)$$

The inverse transformation angle θ from oRGB to $L'C'C'$ is:

$$\theta(\theta_o) = \begin{cases} (2/3)\theta_o & \text{if } \theta_o < \pi/2 \\ \pi/3 + (4/3)(\theta_o - \pi/2) & \text{if } \pi \geq \theta_o \geq \pi/2 \end{cases} \quad (7)$$

These transformations are symmetric for points below the yellow-blue axis. The effect of this rotation along the luma axis on the gamut of the color space can be seen in Figure 9.

3.3 Properties of oRGB Space

oRGB is a simple $R'G'B'$ -based color space that is useful for computational applications, such as color adjustment, color transformation, and color transfer. It retains the notions of hue and saturation

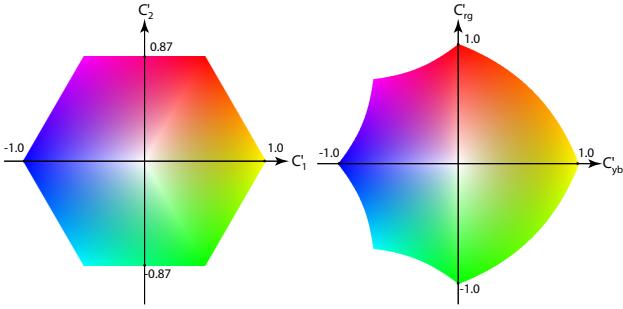


Figure 9: Visualization of the gamut of our color space seen along the luma axis before (left) and after (right) nonuniform rotation. After rotation, the y-axis is a true red-green axis while keeping the yellow-blue the same.

presented in *HSV*, but adds a non-linear perceptual brightness (similar to *HSB*). This similarity to *HSV/HSB* makes *oRGB* a useful space for color picking. Unlike *HSV/HSB*, it encodes perceptual color opponency and its axes encode common color naming used by people [Hardin 1998]. It also provides a natural computational framework for categorizing a color by its color temperature - i.e. warm vs. cool. Finally, the simplicity of the intermediate space $L'C'C'$ allows for straightforward gamut mapping that is critical for computational applications where the manipulations of colors often produces out-of-gamut results.

3.3.1 Locations of Colors

The chroma channels, C'_{yb} and C'_{rg} range in $[-1, 1]$, while the luma channel, L' , ranges in $[0, 1]$. The $[L', 0, 0]$ position on the chroma plane corresponds to a neutral grey color, that is determined by the luma L' . In this constant chroma plane, the hue of a color is its angle in this plane, while its saturation is its distance from the neutral point.

The *oRGB* space places the Hering opponent primaries along the four horizontal and vertical directions. In addition, it places the opponent secondary colors (orange, magenta, cyan, and chartreuse) exactly between the opponent axes, thus creating an intuitive space for color modification based on common color naming (see Figure 10).

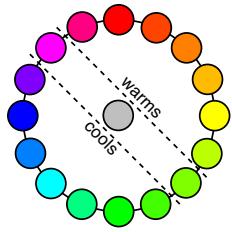


Figure 10: Colors to the right of the magenta-chartreuse axis are warm, to the left are cool, and in the center are temperate.

Studies on basic color naming agree that people across cultures often refer to colors as warm (red or yellow) and cool (blue and green) [Berlin and Kay 1969]. *oRGB* provides a straightforward computational framework for color selection in terms of color temperature. Colors to the right and top of the magenta-chartreuse diagonal (at 45° from the C'_{RG} axis) are warm, while color to the left

and bottom are cool (see Figure 10). Colors towards the center are temperate. We show how this simple categorization can be applied towards cool-to-warm shading models in Section 4.

3.3.2 The *oRGB* Gamut

The gamut of $R'G'B'$ is the cube defined by its orthogonal basis vectors - pure red, green and blue. Linear *RGB* and non-linear $R'G'B'$ spaces have identical gamut - $[0, 1]^3$. When we transform the $R'G'B'$ gamut into $L'C'C'$ space, its (L', C'_1, C'_2) basis vectors are simply the columns of the matrix in Equation 3. The transformed gamut space is thus a parallelepiped. Because we use unequal coefficients for determining luma values, $L'C'C'$ is a true parallelepiped and not a rotated cube like HSV. Since black is $\vec{0}$, its mapping into $L'C'C'$ is also $\vec{0}$, therefore the corner of the gamut cube remains at $\vec{0}$.

Colors that fall within the volume of the transformed gamut space map to in-gamut colors when converted back to $R'G'B'$ space. Colors that fall outside the volume require gamut mapping.

A good gamut mapping method should first try to preserve luminance and then attempt to preserve hue [Stone et al. 1988]. In $L'C'C'$, luminance is a separate axis, so luminance modification and preservation are straightforward. Lines of equal angle also represent the same hue, with distance from the grey point being akin to saturation. These properties allow for simple gamut mapping in $L'C'C'$.

While our preferred editing space uses a further rotation around the luma axis of $L'C'C'$, projecting along lines of equal hue (and therefore equal angle) is equivalent to performing the same projection in *oRGB*. This makes performing gamut mapping for *oRGB* a straightforward, two step process.

First, we ensure that our luma values are in the range of $[0, 1]$. To do so, we generate a piecewise polynomial mapping that ensures that the mean luma value does not move, while values to the right of the mean get compressed to fit within $[\mu, 1]$, and values to the left of the mean fit within $[0, \mu]$:

$$L'(L) = \begin{cases} \mu + (1 - \mu) \left[\frac{(L - \mu)}{(L_{\max} - \mu)} \right]^\beta & \text{if } L > \mu \wedge L_{\max} > 1 \\ \mu - \mu \left[\frac{(L - \mu)}{(L_{\min} - \mu)} \right]^\beta & \text{if } L \leq \mu \wedge L_{\min} < 1 \end{cases}$$

where μ is the average luma, L_{\min} is the minimum luma value, and L_{\max} is the maximum luma value. In practice we have found $\beta = 2/3$ to work fairly well.

Second, we need to determine whether the chroma values are out-of-gamut, and if so how to bring them into gamut. We found two approaches that work well in practice - clamping and scaling.

In clamping, we take any pixel values that are outside the $L'C'C'$ gamut boundary (i.e. the parallelepiped) and project them to the boundary. To do so, we trace a ray from $\langle L'_{pixel}, 0, 0 \rangle$ to the pixel value in question, $\langle L'_{pixel}, C'_{pixel}, C'_{pixel} \rangle$. The ray is traveling along the chroma plane of equal luma for that pixel. If we hit the $L'C'C'$ parallelepiped before we reach the value, we know that the pixel value is outside the gamut space and needs to be clamped to the chroma value found on the intersected boundary. Clamping to the boundary is already a great improvement over first transforming from $L'C'C'$ to $R'G'B'$ and then performing clamping before display (see Figure 11). This is because clamping to the boundary in $L'C'C'$ preserves both the luma and the hue of a particular value, while clamping an $R'G'B'$ triple to $[0, 1]$ does not.

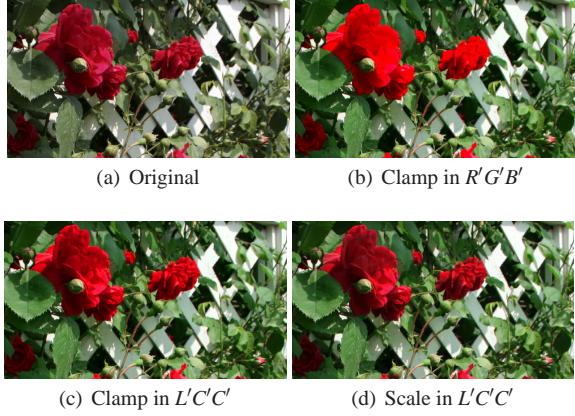


Figure 11: The original image is modified by doubling the standard deviation in the red-green channel resulting in out-of-gamut red values but improved green variation. Clamping in $R'G'B'$ leaves overly saturated reds. Both clamping and scaling in $L'C'C'$ preserve luma, while scaling also produces a more even saturation distribution. Both methods retain the improved green variation.

If many pixel values are outside of the $L'C'C'$ gamut boundary, however, clamping in $L'C'C'$ will produce a very saturated image, as many values will be truncated to the most saturated part of the space. For this case, we scale values towards the center, i.e. $\langle L', 0, 0 \rangle$, instead of simply clamping to the boundary. To do so, we first divide the parallelogram into planes of equal luma. For each equal luma plane, we discretize the plane into many angular slices. For each angular slice, we compute both the furthest distance from the grey point and the closest point on the boundary for all pixel values that fall within the slice. We then project all pixel values within that slice so that the point furthest out will lie on the boundary of the parallelepiped. To avoid color discontinuities we use about 3000 angular slices per luma plane which result in visually smooth mappings of the colors.

Using a simple linear scaling provides excellent results in practice (see Figure 11), though it is possible that some more complicated non-linear method may be necessary in some cases. Gamut mapping via scaling is used for all *oRGB* results in the paper, unless otherwise noted.

4 Sample Applications

We now demonstrate the utility of the *oRGB* color space in four different applications: color adjustment, cool-to-warm shading, color transfer, and gamut mapping. For color adjustment, we show how *oRGB*'s natural opponent axes can be used to provide a meaningful Photoshop-style “variations” tool. As *oRGB*'s natural primaries and secondaries are intuitive they can aid in color selection and can serve as a natural metric for color temperature in cool-to-warm shading. For color transfer, we demonstrate that *oRGB* performs at least as well as previously suggested computational spaces. Finally, we examine gamut mapping in the context of one computational application - color transfer.

4.1 Color Adjustment

Users want to perform simple, yet intuitive color adjustments. Here we show how *oRGB* can be used for Photoshop-style “variations”

as well as for selection of colors for shading models.



Figure 12: Modifying color contrast through deviation adjustment. One step down/left scales deviation by $1/\alpha$ while a step up/right scales deviation in that channel by α . The horizontal axis is a yellow-blue standard deviation change (left lowers and right increases) and the vertical axis is a red-green change (down lowers and up increases).

Our color space is useful for simple color contrast adjustments as we can increase the saturation of the red-green channel independently from the yellow-blue one. To do so, we simply have to scale the standard deviation in each channel separately. This operation provides useful and intuitive variations results (see Figures 12 and 13).

Sometimes, it is more useful to simply take an existing image and make it “warmer” or “cooler” in tone. This operation is straightforward in *oRGB*, as the two chroma channels provide a direct mapping into a color's temperature. Since the two chroma channels are independent, we can adjust them without introducing global image color casts (see Figure 14).

We can also apply the concept of “variations” to material parameters. Varying the diffuse texture via mean shifting in the chroma plane generates meaningful alternative materials (see Figure 15). Shifting up produces a redder tone, up and right - an orange tone, right - a more yellow tone, and right and down - a chartreuse tone. Similarly, shifting left and up produces a magenta tone, left - a bluer tone, left and down - a cyan tone, and down - a greener tone.

4.2 Cool to Warm Shading

Gooch et al. [Gooch et al. 1998] present a cool-to-warm shading model that maps surface normal to color, with normals pointing away from the light being associated with cool colors, and those that point toward the light being warm. They also present a model for choosing the colors based on intrinsic surface color. Their model is:

$$C(\cos) = (1 - f(\cos)) C_{\text{cool}} + f(\cos) C_{\text{warm}}, \quad (8)$$

where \cos is the cosine between surface normal and a light direction, f is a monotonically increasing function that maps $[-1, 1]$ to

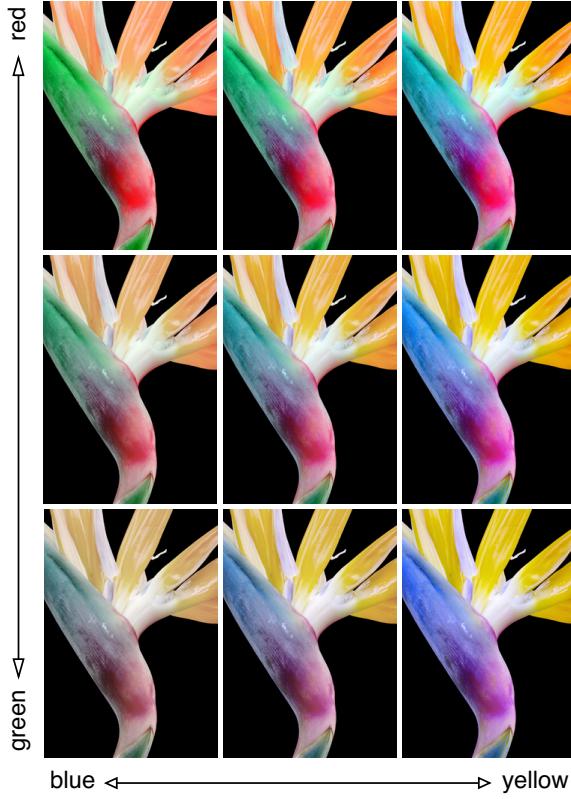


Figure 13: Another example of color adjustment using deviation scaling, similar to Figure 12.

$[0, 1]$, and C_{cool} and C_{warm} are empirically chosen colors. They present results for a linear f and a variety of C_{cool} and C_{warm} .

A difficulty that arises when using this model is that there is no metric that helps users decide how “warm” or “cool” a color is. This issue can be directly addressed in $oRGB$ (see Figure 10), as a color is warm if its dominant direction is toward the red or yellow axes, and cool if its dominant direction is toward the blue or green axes. Quantitatively a warm color is one where $(L', C'_{yb}, C'_{rg}) \cdot (0, 1, 1) > 0$ and a cool color is one where that dot product is negative.

To determine whether the intuitive axes of the $oRGB$ space are helpful in NPR applications, we first added a cool-warm test to an NPR shader that uses Equation 8. We also explored changes to the interpolation rule in an attempt to make NPR shaded objects less flat looking. If the three channels are interpolated separately, then we have:

$$\begin{bmatrix} L' \\ C'_{yb} \\ C'_{rg} \end{bmatrix} = \begin{bmatrix} (1 - f_0(\cos))L^c \\ (1 - f_1(\cos))C^c_{yb} \\ (1 - f_2(\cos))C^c_{rg} \end{bmatrix} + \begin{bmatrix} f_0(\cos)L^w \\ f_1(\cos)C^w_{yb} \\ f_2(\cos)C^w_{rg} \end{bmatrix} \quad (9)$$

where $(L^c, C^c_{yb}, C^c_{rg})$ and $(L^w, C^w_{yb}, C^w_{rg})$ are the $oRGB$ components of C_{cool} and C_{warm} respectively. This will be a valid cool to warm mapping provided the end colors are warm and cool, and that all of f_i are strictly non-decreasing and $f_i(-1) = 0, f_i(1) = 1$. If the two chroma channels are handled as in Gooch et al. [Gooch et al. 1998] and the luma channel is an average between linear interpolation and cosine-based interpolation (Gouraud lighting), then an NPR look can be retained while also having some visual character of diffuse lighting (Figure 16). Our point is not that this is a better

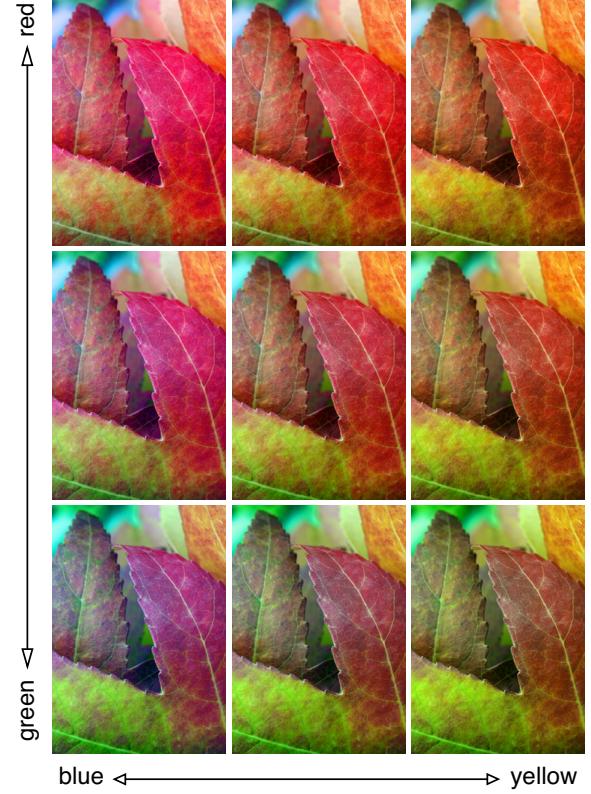


Figure 14: Changing the overall tone of an image through mean shifting. The natural axes of $oRGB$ provide intuitive shifts along primaries (yellow, red, blue, and green) and secondaries (orange, magenta, cyan, and chartreuse). The horizontal axis is yellow (shifting right) and blue (shifting left). The vertical axis is red (shifting up) and green (shifting down).

NPR model, but that $oRGB$ provides a space where reasoning about NPR model design is fairly natural.

4.3 Color Transfer

For color transfer, Reinhard et al. [Reinhard et al. 2001] demonstrated that the $l\alpha\beta$ color space [Ruderman 1998] provides excellent results when compared to the RGB color space. While $l\alpha\beta$ works well in many situations, $oRGB$ provides features we feel are lacking in $l\alpha\beta$. For example, color transfer in $oRGB$ allows for independent manipulation of the luma, pure yellow-blue, and pure red-green channels. Thus, the statistical properties of the channels can be transferred in a space that is meaningful in terms of natural primaries and secondaries.

We compare color transfer results (see Figures 17 and 18) produced by $oRGB$ with those of three different Hering-style opponent color models - *Schwarz* [Schwarz et al. 1987], *RBW* [Shirriff 1993], and *RGBY* [Ware and Cowan 1990]. We show $oRGB$ results that use no gamut mapping or use gamut scaling. In addition, we compare against Adobe Photoshop’s “Color Match” function, as well as HSV , $l\alpha\beta$, $CIE L^*a^*b^*$ and YCC .

In $oRGB$ shifting red towards blue produces purple, while retaining yellow (Figure 17), and shifting magenta towards red results in true red (Figure 18). The opponent spaces we compare against also have orthogonal axes, and the hue shifts in these spaces seem to produce

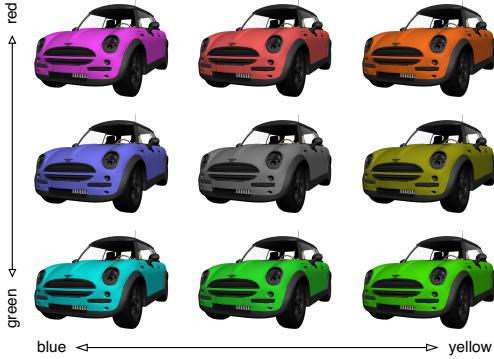


Figure 15: Instead of applying mean shifting to the final image, as in Figure 14, we instead used the intuitive $oRGB$ axes as a color “variations” selector for the diffuse term in a car rendering.

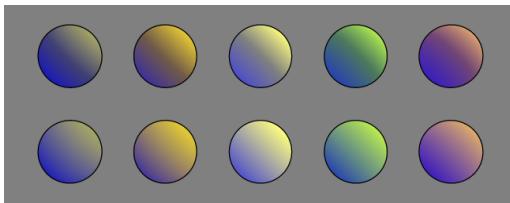


Figure 16: Bottom: shading from Gooch et al. Top: the luma channel in $oRGB$ is modified to partially resemble traditional Gouraud style shading while maintaining a cool-to-warm transition.

reasonable results. However, the saturation and brightness in these spaces do not seem to perform as well, perhaps as a result of the fact that they operate in linear space. In Figure 19 we show the importance of running in non-linear space for $oRGB$. The rest of the spaces do not have orthogonal red-green and yellow-blue axes, and therefore independent shifting and scaling results in mixing some of the color from the other axis (e.g. the magenta cast for $l\alpha\beta$ in Figure 18).



Figure 19: Linear version of our model for color transfer (original and target images are the same as the ones used in 17 and 18).

It should be noted that orthogonality is not the full story for color transfer. RGB has orthogonal axes, but unfortunately for natural images, it has highly correlated axes restricting its use for color transfer [Reinhard et al. 2001]. While $l\alpha\beta$ produces de-correlated axes using PCA, $oRGB$ attempts to utilize opponent-color perceptual axes for meaningful computational applications.

Color transfer is a complex operation that is highly dependent on the specific image distributions. We don’t believe there is one color space that can consistently provide the best results. However, we believe $oRGB$ performs at least as good, and in many cases - better, than other computational color spaces, based on the set of images we’ve tested.

4.4 Gamut Mapping

Gamut mapping is a necessary operation for computational applications, as values are often shifted and scaled, thus increasing the chance for producing out-of-gamut pixel values. Because $oRGB$ is a linear transform of $R'G'B'$ followed by a rotational distortion, the gamut boundary is simpler than for most other spaces. While this does not necessarily guarantee fewer out-of-gamut values, it allows us to correct such cases more easily than in $CIE L^*a^*b^*$ or $l\alpha\beta$.

One computational application, color transfer, works by shifting and scaling groups of pixels without regard to gamut boundaries. While any image used for transferring statistics will have a well defined mean in the RGB gamut space, some values towards the tails of the distributions may land outside the RGB gamut following transfer.

To provide a comparison metric for the out-of-gamut results produced by different color spaces during color transfer, we did the following test. We took the top 12 images produced by Google images, using “family vacation” as a search term. We then did color transfers of the 12 photos to each other, producing 132 non-trivial transfer results per color space. Table 1 summarizes the averaged results for each of the color spaces.

Color Space	% Pixels	% R	% G	% B
$oRGB$	10.36	22.58	16.59	26.42
HSV	9.07	121.87	122.12	118.9
$l\alpha\beta$	10.46	104.54	40.56	25.02
$Schwarz$	16.45	>999.9	>999.9	>999.9
RBW	11.95	146.29	124.20	154.94
$RGBY$	13.67	296.99	234.07	174.70
$CIE L^*a^*b^*$	10.89	117.82	66.05	38.33
YCC	11.24	22.99	15.99	25.91

Table 1: Different color spaces tend to produce different out-of-gamut results. The first column lists the color space tested. The second column shows the average percentage of out-of-gamut pixels. The remaining three columns demonstrate the out-of-gamut percentage range per RGB channel.

$CIE L^*a^*b^*$ and $l\alpha\beta$ produce a similar number of out-of-gamut values to $oRGB$, however, they tend to generate results that are further from the RGB gamut boundary. For these spaces, simply clipping once the color transfer result is returned to RGB can be disastrous. While $oRGB$ has very low out-of-gamut values by comparison, we still seek to remove these remaining out-of-gamut values via gamut mapping.

Having the ability to perform gamut mapping should not be ignored. The $l\alpha\beta$ and $CIE L^*a^*b^*$ results in Figure 17 produce results that are wildly out of the RGB gamut, thus resulting in unpredictable and unexpected color transfer results.

5 Conclusion

The $oRGB$ color space we have introduced has two basic merits. First, it is a relatively simple transformation of $R'G'B'$, so it is efficient, easy to implement, and has a reasonably simple gamut. Second, it has orthogonal axes that are aligned with Hering’s opponent color pairs.

We have shown empirically that this space can be useful in some common computer graphics computations. We suspect the space works well for three reasons. First, the gamma encoding of

(R', G', B') values serves the same role as the non-linearities in models such as $CIE L^*a^*b^*$. Second, the simple gamut boundary reduces out-of-gamut issues and allows any that remain to be easily corrected. Third, we believe that for computer graphics applications, there is something natural about the Hering primaries and their opponency.

There are several limitations of the $oRGB$ space. As a computer graphics space, it is not well-suited to traditional colorimetric applications where all possible colors, as opposed to all displayable colors, are represented. In applications where the Hering primary colors are not central, it is unlikely to have advantages over linear transformations of $R'G'B'$ such as YCC . Perhaps its biggest limitation is that it is not a pure bijection of RGB , so it is not the perfect space for choosing colors for RGB displays.

References

- BERLIN, B., AND KAY, P. 1969. *Basic Color Terms. Their Universality and Evolution*. Berkeley: University of California Press. Reprinted 1991.
- CHICHILISKY, E., AND WANDELL, B. A. 1999. Trichromatic opponent color classification. *Vision Research* 39, 20, 3444–58.
- CONWAY, B. 2002. *Neural Mechanisms of Color Vision*. Kluwer Academic Publishers.
- FAIRCHILD, M. D. 2005. *Color Appearance Models*, 2 ed. Addison-Wesley.
- GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of SIGGRAPH*, 447–452.
- HARDIN, C. L. 1998. Basic color terms and basic color categories. In *Color Vision: Perspectives from Different Disciplines*, W. G. K. Backhaus, R. Kliegl, and J. S. Werner, Eds. De Gruyter, 207–218.
- HUNT, R. 2004. *The Reproduction of Colour*. Wiley.
- HURVICH, L., AND JAMESON, D. 1957. An opponent-process theory of color vision. *Psychol Rev* 64, 384–404.
- KUEHNI, R. G. 2003. *Color Space and its Divisions*. Wiley.
- KUEHNI, R. G. 2004. *Color: an introduction to practice and principles*, second ed. Wiley.
- MOROVIC, J., AND LUO, M. R. 2001. The fundamentals of gamut mapping: A survey. *Journal of Imaging Science and Technology* 45, 3, 283–290.
- NAIMAN, A. C. 1985. Color spaces and color contrast. *The Visual Computer* 1, 3, 194–201.
- POYNTON, C., 2004. Color science and color appearance models for cg, hdtv, and d-cinema. SIGGRAPH course notes.
- REINHARD, E., ASHIKHMEN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *Computer Graphics & Applications*, 2–8.
- RUDERMAN, D. L. 1998. Statistics of cone responses to natural images: Implications for visual coding. *J. Opt. Soc. Am. A* 15, 8 (August), 2036–2045.
- SCHWARZ, M. W., COWAN, W. B., AND BEATTY, J. C. 1987. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. *ACM Trans. Graph.* 6, 2, 123–158.
- SHIRRIFF, K. W. 1993. The RBW color model. *Computers and Graphics* 17, 5, 597–602.
- SMITH, A. R. 1978. Color gamut transform pairs. In *Proceedings of SIGGRAPH*, 12–19.
- STONE, M. C., COWAN, W. B., AND BEATTY, J. C. 1988. Color gamut mapping and the printing of digital color images. *ACM Trans. Graph.* 7, 4, 249–292.
- VALBERG, A. 2005. *Light, vision, color*. Wiley.
- VOLBRECHT, V. J., AND KLIEGL, R. 1998. The perception of blackness: An historical and contemporary review. In *Color Vision: Perspectives from Different Disciplines*, W. G. K. Backhaus, R. Kliegl, and J. S. Werner, Eds. De Gruyter, 187–206.
- WARE, C., AND COWAN, W. 1990. The rgby color geometry. *ACM Trans. Graph.* 9, 2, 226–232.

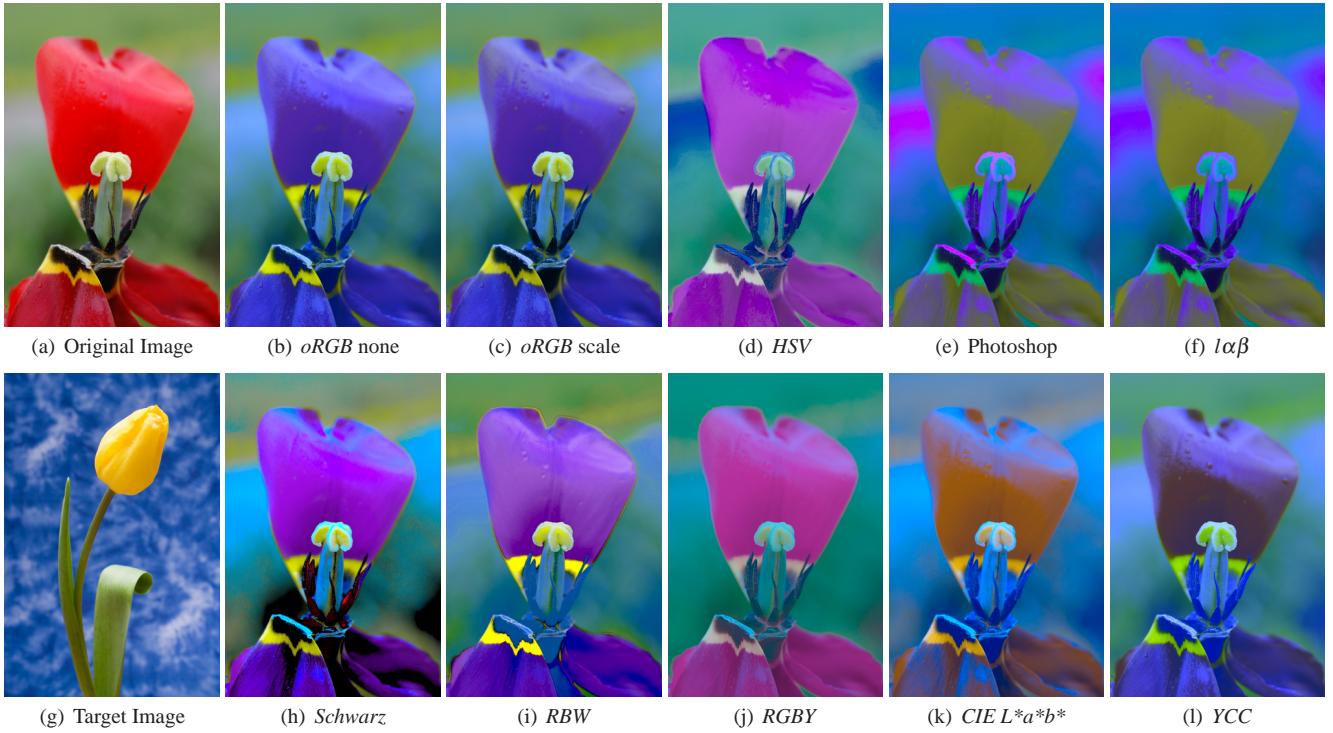


Figure 17: The statistics of the target image are transferred to the original image in a variety of color spaces. Transferring red towards blue produces purple in *oRGB* instead of yellows, reds, or browns produced by the other spaces.

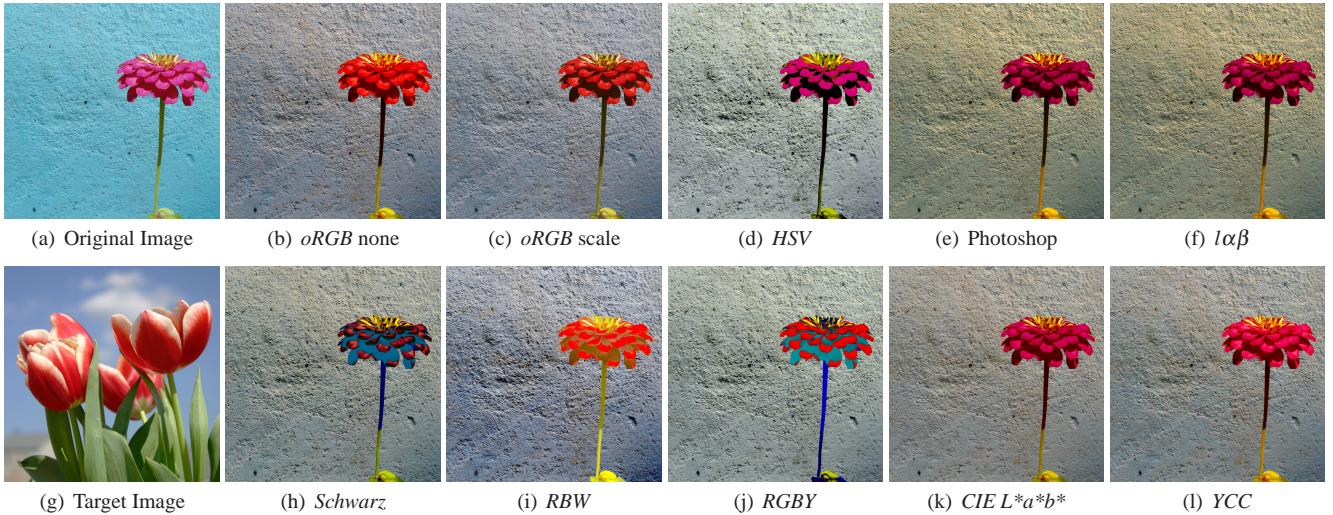


Figure 18: The statistics of the target image are transferred to the original image in a variety of color spaces. A magenta shifted towards pure red becomes pure red in *oRGB* instead of remaining magenta.