GaTech CS-6497 Fall 2019 Project 1: Third Color Selection


Russell Strauss


Jiewen Wang

1. Project Overview

In this project, we developed a web-based application for color matching and 3D color space visualization. User may select two colors and choose one of four color-matching methods: Blend, Background, Accent and Triad. The application automatically generates a color and visualize the triad of colors in RGB or HSL space. Additionally, user can preview the output color under different brightness to see if the result remains a good match when environment lighting changes.

All relevant code is contained within [assets\js\components\scene.js](). Detailed installation guide can be found within https://github.com/russellstrauss/p1-color

2. Blended Color

We blend two colors by taking their mid-point in Lab(CIELAB) color space. In Lab space, L is for lightness, a and b are for chromaticity. It is designed so that the same amount of numerical change in L/a/b values correspond to same amount of visually perceived range.[i] The blended color is perceptually equidistant to two input colors.

Prior to converting from RGB to Lab space, we transform to XYZ color space firstly. It is a linear transformation as following.[ii]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{b_{21}} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{0.176\,97} \begin{bmatrix} 0.490\,00 & 0.310\,00 & 0.200\,00 \\ 0.176\,97 & 0.812\,40 & 0.010\,63 \\ 0.000\,00 & 0.010\,00 & 0.990\,00 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Conversion from XYZ to Lab color space is non-linear.

$$\begin{aligned} L^\star &= 116\, f\left(\frac{Y}{Y_\mathrm{n}}\right) - 16 \\ a^\star &= 500 \left(f\left(\frac{X}{X_\mathrm{n}}\right) - f\left(\frac{Y}{Y_\mathrm{n}}\right) \right) \\ b^\star &= 200 \left(f\left(\frac{Y}{Y_\mathrm{n}}\right) - f\left(\frac{Z}{Z_\mathrm{n}}\right) \right) \end{aligned} \quad \text{in which} \quad f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}$$
$$\delta = \frac{6}{29}$$

And we choose the parameters as: $X_n = 95.0489, Y_n = 100, Z_n = 108.884$

The reverse transformation is:

$$\begin{aligned} X &= X_\mathrm{n} f^{-1}\left(\frac{L^\star + 16}{116} + \frac{a^\star}{500}\right) \\ Y &= Y_\mathrm{n} f^{-1}\left(\frac{L^\star + 16}{116}\right) \\ Z &= Z_\mathrm{n} f^{-1}\left(\frac{L^\star + 16}{116} - \frac{b^\star}{200}\right) \end{aligned} \quad \text{in which} \quad f^{-1}(t) = \begin{cases} t^3 & \text{if } t > \delta \\ 3\delta^2 \left(t - \frac{4}{29}\right) & \text{otherwise} \end{cases}$$

Finally, we convert from XYZ to RGB color space.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.418\,47 & -0.158\,66 & -0.082\,835 \\ -0.091\,169 & 0.252\,43 & 0.015\,708 \\ 0.000\,920\,90 & -0.002\,549\,8 & 0.178\,60 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

ColorInput1 + ColorInput 2 = BlendedOutput

- **The formation is (usually) invariant when brightness is changed slightly**

  We change the brightness of a color by multiplying R, G, B values by a certain factor s. New values $(r', g', b') = (sr, sg, sb)$. This transformation is the same as changing exposure in photo editing software like Photoshop.[iii]

  Conversion between RGB and XYZ color space are linear transforms. Given 2 RGB colors$(r_1, g_1, b_1)(r_2, g_2, b_2)$, their midpoint in XYZ space will still be the midpoint when overall brightness is changed. We can show that, in most cases, converting the midpoint in XYZ space to Lab space does not change the midpoint property, even if this conversion is nonlinear.

  When evaluating $f(t), t > \delta^3$ means that $X/X_n, Y/Y_n, Z/Z_n$are greater than 0.0088. The factors X_n, Y_n, Z_n are approximately 100, so X, Y, Z are greater than 0.88. This means $ir + jg + kb$, a linear combination of (r, g, b), is greater than 0.157 by substituting into RGB to XYZ conversion formula. The factors i, j, k are given by three rows of transformation matrix $\{b_{mn}\}$.

  $(ir + jg + kb) > 0.157$ is approximately the same as $(r + g + b)/3 > 0.157$, given that the summation of each row in matrix $(i + j + k)$ is the summation of each row in matrix $\{b_{mn}\}$, and it is always 1.

  0.157 is a very low value in RGB space. If $r = g = b = 0.157$, the color looks like this ■. Nearly black. We can safely assert that in most situations, especially in natural objects and mixed pigments, R/G/B component of a color is larger than 0.157.

  Once we have this condition, $f^{-1}(t) = t^{1/3}$ holds true in most cases. If we multiply each of X/Y/Z component by s, $f(st) = s^{1/3} f(t)$ (t denotes any of $X/X_n, Y/Y_n, Z/Z_n$)

  Therefore, after the brightness is changed, new Lab color (L', a', b') is

$$\begin{cases} L' = 116 s^{1/3} f(Y/Y_n) - 16 \\ a' = 500 s^{1/3} a \\ b' = 200 s^{1/3} b \end{cases}$$

  If (L_3, a_3, b_3) is the midpoint of (L_1, a_1, b_1) and (L_2, a_2, b_2), the new color with changed brightness $(L'_3, a'_3, b'_3)$is still the midpoint of (L'_1, a'_1, b'_1) and (L'_2, a'_2, b'_2)

ColorInput1 + ColorInput 2 = BackgroundOutput

3. Background Color

We want a color that will contrast well with both input colors. To do this, we first get a blend of the two colors by calculating the midpoint of each point in RGB color space, as laid out above. This will give us a color that is equidistant in hue from both colors which satisfies the condition that the output color will work equally well with both inputs. Our second condition is that the background color will contrast well with the two input colors. For this, we calculate the complementary color of the blended color. The complementary color is defined as the color opposite on the color wheel. So we can convert our RGB color to HSL in order to manipulate the hue to find our complementary color. So we rotate the point about the color wheel by π:

var outputColor = blendedColor.offsetHSL(.5, 0, 0);

Normally the hue value is calculated with either degrees or radians, but our system uses a scale of 0...1. By calculating the linear interpolation from 0...1 to 0...2π, we know that the rotation of π is simply .5, so we perform this transformation on the hue value.

4. Accent Color

In this task, two input colors are specified in HSL (Hue, Saturation, Luminance) color space as $(h_1, s_1, l_1), (h_2, s_2, l_2)$. Each value lies in the range 0…1. The goal is to generate an accent color that contrasts equally with two input colors. The contrast is achieved in three ways:

1. Hue contrast

   By picking a color following "split complementary" rule[iv], we can create maximum and equal contrast with two input colors. We take the average hue of two colors and add 0.5 to it to get new hue value, which is located on the opposite side of two colors.

Split Complementary

2. Saturation contrast

   When input colors are unsaturated, the contrast between them is not high enough. We choose to pick an accent color with higher contrast in saturation. We use an empirical criterion that if the

average saturation of two color is smaller than 0.3, the output color's saturation is increased to $(s_1 + s_2)/2 + 0.5$. Otherwise, the output saturation is the average saturation when both input colors are adequately saturated, because there is already enough contrast.

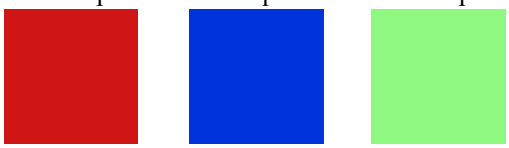
3. Luminance contrast

We start from the intuition that, when two colors are both very bright/dark, an effective way to contrast with them (and as equally as possible) is to use a darker/brighter color. When one is dark and the other is bright, we might just use a color with average luminance to contrast. The output luminance is formulated as:

$$l_3 = \begin{cases} \frac{l_1 + l_2}{2} - \Delta l, if\ |l_1 - l_2| < l_0\ and\ \frac{l_1 + l_2}{2} > 1 - 0.5l_0 - \Delta l \\ \frac{l_1 + l_2}{2} + \Delta l, if\ |l_1 - l_2| < l_0\ and\ \frac{l_1 + l_2}{2} < 1 - 0.5l_0 - \Delta l \\ \frac{l_1 + l_2}{2}, if\ |l_1 - l_2| > l_0 \end{cases}$$

l_0 is the threshold that represents how far apart two colors are in brightness. Δl is luminance shift to create a larger contrast. The choices of these two variables are subjective. In our current solution, we set $l_0 = 0.2$ and $\Delta l = 0.3$

ColorInput 1 + ColorInput 2 = AccentOutput



- **The formation is valid when brightness is changed slightly**

Above formations of hue and saturation are independent of luminance. We claim that high luminance contrast is preserved when luminance of all three colors l_1, l_2, l_3 are scaled and become $(l'_1, l'_2, l'_3) = (sl_1, sl_2, sl_3)$

$$sl_3 = \begin{cases} \frac{sl_1 + sl_2}{2} - s\Delta l = \frac{l'_1 + l'_2}{2} - \Delta l' \\ \frac{sl_1 + sl_2}{2} + s\Delta l = \frac{l'_1 + l'_2}{2} + \Delta l' \\ \frac{sl_1 + sl_2}{2} = \frac{l'_1 + l'_2}{2} \end{cases}$$

The ideal l_3^* is calculated as,

$$l_3^* = \begin{cases} \frac{l'_1 + l'_2}{2} - \Delta l \\ \frac{l'_1 + l'_2}{2} + \Delta l \\ \frac{l'_1 + l'_2}{2} \end{cases}$$

The only difference between our output and ideal value is that Δl becomes $\Delta l' = s\Delta l$. In fact, Δl is chosen subjectively and should be allowed to vary in a certain range. As long as the luminance change s is small, Δl and $s\Delta l$ are within the acceptable range.

5. Triad Color

For our triad color, we need to determine a third output color that cannot be distinguished from the other two. For this, we continue to use our HSL color model to perform the calculations and transformations. The first step is to calculate the hue values of both color inputs and measure the angle between them in the HSL space. Since the angle on the color wheel represents the hue, we simply measure that angle and add a third color either in the positive or negative direction from one of the colors. Then the third color will be distinct from the other two colors since all three will be equidistant from each other around the circle. Very similar to our background color above, this equation is:

triad.offsetHSL(hsl2.h + angle, 0, 0);

ColorInput 1 + ColorInput 2 = TriadOutput

6. **Dimming the lights**

Part of the requirement included performing all the same tasks above, but to also perform these transformations with dimmed or brightened light values. Instead of modifying our original color calculation algorithms, we decided to leave them in place and simply scale the luminance values of the selected and output colors to prevent any room for errors in the algorithms. This lets us abstract out the dimming value calculations from the color output calculations which will also allow us to perform these transformations on any future algorithms without extra code or math. As you can see in the HSL color space, the center disc contains all of the perfectly saturated colors, losing saturation as you approach the center of the circle and changing hues as you move around the outside of the circle. As you can see while scaling the luminance, all colors converge to black in the bottom cone approaching 0% luminance and converge perfectly to white in the top cone approaching 100% luminance. The RGB Cube model functions similarly with black in one far corner of the cube and white in the opposite corner. So scaling the luminance in that color space will send colors from one end of the cube to the other, through the diagonal. The cube diagonal forms every hue-less version of gray in the RGB space.

ColorInput1 * LuminanceScaling + ColorInput2 * LuminanceScaling = BlendedColorOutput * LuminanceScaling

Reference

[i] CIELAB color space, Wikipedia, https://en.wikipedia.org/wiki/CIELAB_color_space
[ii] CIE 1931 color space, Wikipedia, https://en.wikipedia.org/wiki/CIE_1931_color_space
[iii] What is the math behind exposure adjustment on photoshop? Stack Overflow, https://stackoverflow.com/questions/12166117/what-is-the-math-behind-exposure-adjustment-on-photoshop
[iv] Illustrator's Colour Harmonies, Tony Harmer's Design Jungle, Adobe Design Products Resource, https://blogs.adobe.com/tonyharmer/2015/03/23/illustrators-colour-harmonies/