Differentiable Color Mixing

INAN XU, UC San Diego, USA

This project attempts to simulate a real-world color mixing process. The initial goal was given a mixed color, identify the two underlying colors used to mix it. This goal was accomplished in RGB additive color space. To simulate a real-world color mixing process, I learned the absorption and scattering curves from Kubelka-Munk color theory using automatic differentiation to discover the mixed colors.

Fig. 1. A gradient of two mixed colors.

1 INTRODUCTION

Colors are defined by color spaces: some mathematical representations that define the space of possible colors. There are countless color spaces with various pros and cons. One common digital color space is RGB, where colors are represented by their red (R), green (G), and blue (B) components and additively mixed to achieve the final color. Of course, colors in the physical world cannot use such simple representations. One possible representation is Kubelka-Munk.

Kubelka-Munk color theory represents mixing of colors using their light absorption and scattering properties. Specifically, a color is defined by two curves $K \in [0, 1]$ and $S \in [0, 1]$ across the visible light spectrum. For each wavelength of color, the amount of light absorbed and scattered varies, thus producing a different color. We can mix two colors together using a mixing ratio t:

$$C_{mix}(\lambda) = C_1(\lambda) \cdot (1 - t) + C_2(\lambda) \cdot t \tag{1}$$

Kubelka-Munk is more appealing for practical applications because it better matches real-world color mixing processes than digital color spaces. From this, we could theoretically discover the absorption and scattering curves to mix two colors together to acheive the target color.

In this report, I present a program written in Loma that tries to discover the absorption and scattering curves such that the two colors mix together to match the target color. I also implemented a simpler version of RGB color mixing as a proof-of-concept.

2 METHOD

The core approach is to model the color mixing process in Loma, then differentiate it to use gradient descent to discover the input parameters.

For simple RGB color mixing, two colors can be mixed as follows. Given $C_1 = (r_1, g_1, b_1)$ and $C_2 = (r_2, g_2, b_2)$ where all values are normalized to the range $[0, 1]$, a mixed color is taking the element-wise average of each color component:

$$C_{mix} = (\frac{1}{2}(r_1 + r_2), \frac{1}{2}(g_1 + g_2), \frac{1}{2}(b_1 + b_2)) \tag{2}$$

For Kubelka-Munk color mixing, the approach is roughly taken from Sochorova et al. [1] We define each color by a ratio of its absorption and scattering curves along the visible light spectrum:

Author's address: Inan Xu, inxu@ucsd.edu, UC San Diego, San Diego, California, USA.

$C = K(\lambda_i)/S(\lambda_i)$. We mix the two colors element-wise by wavelength to get the mixed color C_{mix} by the equation described earlier.

To visualize this color, we need to convert it to RGB color space. We define a reflectance and Saunderson function to model real-world light-color properties.

$$R(\lambda) = 1 + C(\lambda) - \sqrt{C(\lambda)^2 + 2C(\lambda)}$$
$$S(\lambda) = \frac{(1 - K_1)(1 - K_2)R(\lambda)}{1 - K_2 R(\lambda)}$$

where K_1 and K_2 are the internal and external reflection coefficients respectively. From here, we convert this to XYZ color space and then RGB color space through a series of linear transformations.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{M}{Y_{D65}} \int_\lambda S(\lambda) \begin{bmatrix} \tilde{x}(\lambda) \\ \tilde{y}(\lambda) \\ \tilde{z}(\lambda) \end{bmatrix} d\lambda \tag{3}$$

where Y_{D65} is a normalization factor that approximates natural light, M is a matrix of constants to convert from XYZ to RGB color space where $M \approx \begin{bmatrix} 3.24 & -1.53 & -0.49 \\ -0.96 & 1.87 & 0.04 \\ 0.05 & -0.20 & 1.05 \end{bmatrix}$, and $\begin{bmatrix} \tilde{x}(\lambda) \\ \tilde{y}(\lambda) \\ \tilde{z}(\lambda) \end{bmatrix}$ are CIE 1931 color matching functions to convert to XYZ.

2.1 Implementation

Simple RGB color mixing is implemented as described earlier. For Kubelka-Munk, I made various simplifications. First, the number of wavelengths along the visible spectrum is far too many, so I estimated it using a fixed number of bands. Additionally, the Y_{D65}, M, and other various constants are pre-computed and approximated using values found online. The gradient descent method for both is similar to examples given in the homeworks, and various visualizations were similarly implemented using matplotlib.

The code can be found on GitHub at `github.com/ixukw/diff_color_mixing`.

2.2 Loma Restrictions

I ran into a few restrictions during implementation in Loma. First, there is no way to define a list in Loma. For example, the following statement `x:Array[float]=[1,2,3]` is illegal. Thus, when defining my pre-computed constants, each index had to be defined manually. I also could not come up with a way to pass them from the python program, as they were required during the forward differentiation call.

As a result, my performance for Kubelka-Munk color theory is lacking as I restricted the number of wavelength bands to 5. Inputting a 38 by 3 2D array would have been far too time consuming.

3 EXPERIMENTAL RESULTS

Let's start with the simple RGB results. Figure 1 showcases a sample run. The color initially starts at a random value, and converges to the target color after a number of iterations. We observe a typical smooth gradient along all three axes for red, green, and blue components.

Next, let's see the current results for color mixing in Kubelka-Munk color space. Figure 2 contains a plot with the loss during gradient descent. Clearly, the results are not exactly correct. First, this is likely because there are only five wavelength bands; normally there would be 300 as the visible spectrum is from 400nm to 700nm. Second, there may be a logic error where certain gradients were not being updated properly. The gradient for each index in the absorption and scattering curves

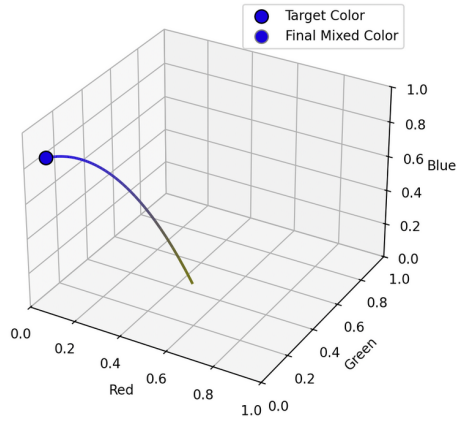

Fig. 2. Gradient descent for RGB color mixing.

must be updated with each call, but I noticed values were failing to be preserved across calls. With more time, I could likely resolve this issue.

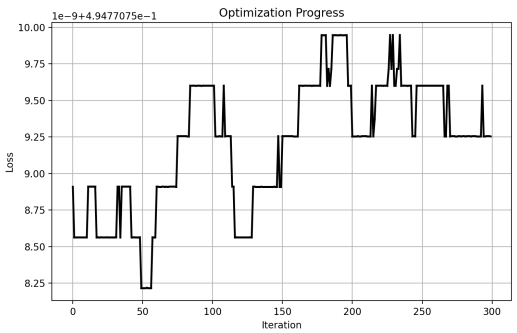

Fig. 3. K/S curve loss for Kubelka-Munk color mixing. The line should display the color, but due to issues the color is black throughout.

With more wavelength bands, the loss issues would likely be resolved and gradient descent could actually converge to the desired color. When the number of wavelength bands is sparse, the curve becomes increasingly sparse thus affecting the final result.

4 TAKEAWAYS AND FUTURE WORK

Accurately modeling color spaces is difficult. There are a variety of parameters to include and physics equations to model. Differentiable programming allows such physics equations to be both easily modeled and accurately modeled, compared to neural networks which rely on approximations. An area I wanted to explore further was actually modeling light diffraction, but ran out of time.

4.1 Optimizer

Simple gradient descent may not be sufficient for more complex color representations. For RGB, six parameters was sufficient to converge gradient descent to the correct optimum. Kubelka-Munk

requires at least 30 wavelength bands to perform well, which is already 120 parameters which may not be possible to learn. A simple neural network may be required to learn such representations.

4.2 Future Direction

Practically, modeling color is beneficial for digital artists and painting machines. Simulating color in complex environments and how various pigments mix based on their composition is a difficult task. In particular, the discrepancy between digital colors and physical colors is an important factor in several fields like design. Furthermore, there exists a difference between color and chromaticity: the former of which involves brightness, another factor that could be interesting to look at.

REFERENCES

[1] Šárka Sochorová and Ondřej Jamriška. 2021. Practical pigment mixing for digital painting. *ACM Trans. Graph.* 40, 6, Article 234 (Dec. 2021), 11 pages. https://doi.org/10.1145/3478513.3480549