Hue Preserving Minimum Color Difference Gamut Clipping (HPMINDE)

Shankhya Debnath

August 22, 2024

1 Introduction

Hue Preserving Minimum Color Difference (HPMINDE) gamut clipping is an algorithm designed to map colors from one color space to another while preserving the hue and minimizing the perceptual color difference. This document details the mathematical background of the HPMINDE process, provides the implementation of the algorithm in Python, and discusses the results and observations.

2 Mathematical Background

The primary goal of the HPMINDE algorithm is to clip colors that fall outside the target gamut boundary while preserving the hue and minimizing the perceptual color difference. The process can be broken down into the following steps:

2.1 Color Representation in CIELAB Space

The CIELAB color space is designed to be perceptually uniform, meaning that a given numerical change in a color value should correspond to roughly the same perceptual change in color. A color in this space is represented as a three-dimensional vector:

$$\mathbf{L}^*\mathbf{a}^*\mathbf{b}^* = \begin{bmatrix} L^* \\ a^* \\ b^* \end{bmatrix} \tag{1}$$

where L^* represents lightness, a^* represents the position between red/magenta and green, and b^* represents the position between yellow and blue.

2.2 Chroma and Hue Angle

Chroma (C^*) and hue angle (h) are derived from the a^* and b^* components:

$$C^* = \sqrt{a^{*2} + b^{*2}} \tag{2}$$

$$h = \arctan\left(\frac{b^*}{a^*}\right) \tag{3}$$

2.3 Hue Preserving Minimum Color Difference

Given a color in the LAB space and the Gamut Boundary Descriptor (GBD) obtained via a convex hull, the HPMINDE algorithm finds the closest point on the GBD while preserving the hue. The weighted color difference (ΔE_{hp}) is calculated as:

$$\Delta E_{hp} = \sqrt{(L_1^* - L_2^*)^2 + (C_1^* - C_2^*)^2 + \left[(h_1 - h_2) \times C_1^*\right]^2} \tag{4}$$

The hue difference $(h_1 - h_2)$ is weighted by the chroma C_1^* to ensure that larger chroma values result in greater emphasis on hue preservation. The algorithm iterates through the GBD points and selects the point that minimizes ΔE_{hp}, ensuring that the hue is preserved while minimizing the color difference.

2.4 Convex Hull for Gamut Boundary Descriptor (GBD)

To represent the gamut boundary, the LAB points corresponding to the AdobeRGB gamut are used to compute a convex hull. The convex hull represents the minimal bounding surface that encloses all the LAB points:

$$\mathbf{GBD} = \text{ConvexHull}\left(\mathbf{L}^*\mathbf{a}^*\mathbf{b}^*\right) \tag{5}$$

This convex hull is used to determine whether a color falls inside or outside the gamut. If a color is outside the gamut, the closest point on the convex hull is found using the HPMINDE algorithm.

3 Implementation

The HPMINDE (Hue Preserving Minimum Color Difference) algorithm is implemented to perform gamut mapping from the sRGB color space to the AdobeRGB color space. The process begins with converting the input image from the sRGB color space to the CIELAB color space, which is used for its perceptual uniformity. LAB points corresponding to the AdobeRGB color gamut are generated using an ICC profile, and a convex hull is applied to these points to create the Gamut Boundary Descriptor (GBD). The convex hull defines the boundary within which all AdobeRGB colors reside in LAB space.

The HPMINDE algorithm then maps each pixel's color in the image to the closest point within the GBD while preserving the hue. This is achieved by

calculating the chroma and hue of the original color and comparing it to the points on the GBD. The color is then adjusted to minimize the weighted color difference while maintaining hue consistency. The result is a color-mapped image that preserves the perceptual attributes of the original image while fitting within the AdobeRGB gamut. The implementation also includes visual comparisons of the original and mapped gamuts in the L/C plane and the original and mapped images side by side, highlighting the effectiveness of the gamut mapping process.

4 Observations

The HPMINDE algorithm effectively preserves the hue of the original colors while mapping them within the target gamut, as observed in both the resulting images and gamut plots. The algorithm minimizes the perceptual color difference (ΔE), ensuring that the clipped colors are as close as possible to their original values in terms of perceived color. Although effective, the algorithm's use of convex hull computation and point-by-point color comparison can be computationally expensive, especially for high-resolution images.

5 References

1. Farup, I., Hardeberg, J. Y., & Amsrud, M. (2004, January). Enhancing the SGCK colour gamut mapping algorithm. In Conference on Colour in Graphics, Imaging, and Vision (Vol. 2, pp. 520-524). Society of Imaging Science and Technology.

2. Kolås, Ø., & Farup, I. (2007, January). Efficient hue-preserving and edge-preserving spatial color gamut mapping. In Color and Imaging Conference (Vol. 15, pp. 207-212). Society of Imaging Science and Technology.

3. Morovič, J. (2008). Color gamut mapping. John Wiley & Sons.