https://doi.org/10.2352/EI.2025.37.10.IPAS-226

Computationally-Efficient Hue-Preserving Gamut Mapping in RGB and YUV

Touraj Tajbakhsh, Meta
Hamid Mirzaei, Meta

Abstract

In this paper, we present a computationally-efficient gamut mapping algorithm designed for tone-mapped images, focusing on preserving hue fidelity while providing flexibility to retain either luminance or saturation for visually consistent results. The algorithm operates in both RGB and YUV color spaces, enabling practical implementation in hardware and software for real-time systems. We demonstrate that the proposed method effectively mitigates hue shifts during gamut mapping, offering a computationally viable alternative to more complex methods based on perceptually uniform color spaces.

Introduction

Gamut mapping is a well-studied problem in digital imaging [1], important for ensuring that colors remain consistent when images undergo tone mapping and other transformations. However, existing methods often rely on complex operations in perceptually uniform color spaces such as CIELAB or IPT [2], which can be computationally intensive and challenging to implement in hardware.

In contrast, we propose a simple yet effective gamut mapping algorithm that works directly in RGB and YUV spaces. This method is particularly suitable for applications where computational efficiency is paramount, such as real-time image processing in digital videography.

We frame the problem in the context of tone mapping, where operations such as tone curves or detail layers can modify the luminance component of an image in linear color spaces. However, after tone mapping, the resulting values are often transformed into a non-linear color space, leading to issues such as out-of-gamut colors and unwanted hue shifts, especially when dealing with high dynamic range (HDR) content. While our approach is particularly effective in addressing these challenges within tone mapping, it is not limited to this application. The proposed solution is applicable whenever there is a need to reduce the gamut of an RGB or YUV color space to fit into a canonical space, using the original unmodified RGB values as hue priors for the mapping.

Problem Statement and Approach

Tone mapping is the process of compressing the dynamic range of a luminance image Y_0 to achieve a desired luminance Y_{tm}. This process involves applying operations such as tone curves to adjust the brightness and contrast of the image. The relative luminance Y_0 is often derived from the input RGB triplet $\mathbf{m}_0 = (R_0, G_0, B_0)$ using the equation:

$$Y_0 = \lambda \mathbf{c} \cdot \mathbf{m}_0^T + (1-\lambda)\max(R_0, G_0, B_0) \tag{1}$$

Here, $\mathbf{c}$ is a coefficient vector (e.g., from Rec. 709), and λ is a blending factor that balances between a weighted sum of the RGB channels and the maximum channel value [3].

While tone mapping adjusts the luminance component, the challenge lies in reproducing the colors in a way that maintains fidelity to the original image. This is where tone reproduction techniques come into play. Schlick [6] proposed a method to map the RGB values of the tone-mapped image as follows:

$$\mathbf{m}_{\text{tm}} = \left(\frac{\mathbf{m}_0}{Y_0}\right)^p Y_{\text{tm}} \tag{2}$$

Alternatively, Mantiuk et al. [7] suggested a different approach:

$$\mathbf{m}_{\text{tm}} = \left(\left(\frac{\mathbf{m}_0}{Y_0} - 1.0\right)^p + 1.0\right) Y_{\text{tm}} \tag{3}$$

The parameter p controls the degree of saturation adjustment in the tone-mapped image. These methods and others are discussed in the broader context of tone reproduction in the work by Pouli et al. [4].

However, these tone reproduction techniques can introduce two key issues:

i. **Out-of-Gamut Colors:** The tone-mapped RGB values $\mathbf{m}_{\text{tm}}$ can exceed the displayable range $[0,1]$, even if the original values were within bounds.

ii. **Hue Shifts:** The non-linear saturation changes in the Schlick and Mantiuk formulations can cause hue shifts, particularly when different color channels are affected unequally.

To address the out-of-gamut color issue, the tone-mapped values are typically clipped to the displayable range:

$$\mathbf{m}_{\text{clip}} = \min(1, \mathbf{m}_{\text{tm}}) \tag{4}$$

At this point, it's important to clarify that in the context of the gamut mapping algorithm, we refer to "luma" instead of "luminance" because we assume that the incoming RGB or YUV values have been transformed into a gamma-corrected or otherwise non-linear color space. Earlier in this section's discussion, we referred to luminance in the context of tone reproduction operators, which typically operate on linear RGB values. However, for the purposes of gamut mapping, $\mathbf{m}_0$ and $\mathbf{m}_{\text{clip}}$ are assumed to be transformed into a non-linear space prior to the application of the gamut mapping algorithm. To maintain readability, we have avoided adding additional notations to these variable names, but it is understood that they represent values in a non-linear color space for the gamut mapping process.

Clipping is the simplest form of gamut mapping; however, it can exacerbate hue shifts, necessitating a more sophisticated approach. Our proposed method aims to improve tone reproduction by preserving hue while controlling either luma or saturation. When preserving hue with luma, there must be a trade-off with saturation, and vice versa.

Our method demonstrates that with minor adjustments, gamut mapping can retain either the full luma or the full saturation of $\mathbf{m}_{\text{clip}}$ while preserving the original hue of $\mathbf{m}_0$. By introducing a blending weight, we can also balance the trade-off between saturation and luma preservation, leading to a more flexible and balanced approach to gamut mapping.

When aiming for full luma or saturation retention, we target the qualities of $\mathbf{m}_{\text{clip}}$ because the tone-mapped and clipped image, brought into the displayable range, typically exhibits the most highlight contrast and saturation, albeit often with significant hue shifts. Our method corrects the hue shifts while retaining the desired luma or saturation, or a blend of both, to achieve a more visually consistent result.

Gamut Mapping in RGB Color Space

The proposed gamut mapping algorithm in the RGB color space is designed to preserve the original hue h_0 and adjust either luma or saturation, depending on the chosen formulation, to retain the desired attribute of $\mathbf{m}_0$ and $\mathbf{m}_{\text{clip}}$.

1. **Hue Preservation:** The original hue, h_0, is defined by the cross-color channel ratio:

$$h_0 = \frac{R_0 - G_0}{B_0 - G_0}, \tag{5}$$

   and the final output RGB values $\mathbf{m}_{\text{out}}$ adopt the hue h_0 of the original unmodified $\mathbf{m}_0$ after gamut mapping.

2. **Luma or Saturation Retention:** To retain luma, the output RGB values are adjusted to match the luma of $\mathbf{m}_{\text{clip}}$. This is achieved by calculating two intermediate luma values, V_0 and V_{clip}, as follows:

$$V_0 = \mathbf{c} \cdot \mathbf{m}_0^T \quad \text{and} \quad V_{\text{clip}} = \mathbf{c} \cdot \mathbf{m}_{\text{clip}}^T, \tag{6}$$

   where $\mathbf{c}$ is a vector of coefficients (e.g., from Rec. 709). This approach ensures that the output image maintains luma consistency with the clipped image, $\mathbf{m}_{\text{clip}}$, but may introduce some desaturation to ensure that all colors remain within the displayable gamut.

   Alternatively, to prioritize saturation retention, the intermediate luma values V_0 and V_{clip} can be defined as:

$$V_0 = \min(\mathbf{m}_0) \quad \text{and} \quad V_{\text{clip}} = \min(\mathbf{m}_{\text{clip}}). \tag{7}$$

   This approach retains the saturation of $\mathbf{m}_{\text{clip}}$ but may cause the luma to decrease, particularly in areas of high brightness.

3. **Blended Luma and Saturation Retention:** To provide a flexible balance between luma and saturation retention, a weighted formulation can be used to blend between the two. We introduce a blending weight w, where $w \in [0,1]$. When $w = 1$, the method prioritizes luma retention, and when $w = 0$, it prioritizes saturation retention. Two baseline values V_0 and V_{clip} are then defined as:

$$V_0 = w \cdot (\mathbf{c} \cdot \mathbf{m}_0^T) + (1-w) \cdot \min(\mathbf{m}_0), \tag{8}$$

$$V_{\text{clip}} = w \cdot (\mathbf{c} \cdot \mathbf{m}_{\text{clip}}^T) + (1-w) \cdot \min(\mathbf{m}_{\text{clip}}), \tag{9}$$

   allowing for a smooth transition between luma retention and saturation retention based on the content and priorities of the image.

The algorithm then calculates a gain factor g based on the maximum RGB values before tone mapping and after tone mapping and clipping:

$$g = \frac{\max(R_{\text{clip}}, G_{\text{clip}}, B_{\text{clip}}) - V_{\text{clip}}}{\max(R_0, G_0, B_0) - V_0}. \tag{10}$$

Here, V_0 and V_{clip} act as baselines around which the color channels are scaled. Their definitions (luma or minimum channel) determine whether the method emphasizes luminance or saturation preservation. The term $\max(R_0, G_0, B_0) - V_0$ represents how far the original color's maximum channel sits above its baseline, while $\max(R_{\text{clip}}, G_{\text{clip}}, B_{\text{clip}}) - V_{\text{clip}}$ is the analogous "height" in the clipped image. Taking the ratio of these two heights produces a single global scale factor g. This ensures that, after scaling, the largest channel in the final image remains at or below its clipped maximum, preventing new out-of-gamut values.

The final output RGB values $\mathbf{m}_{\text{out}}$ are then determined by applying g to the channel differences relative to V_0:

$$\mathbf{m}_{\text{out}} = V_{\text{clip}} + \left(\mathbf{m}_0 - V_0\right) \times g. \tag{11}$$

The output $\mathbf{m}_{\text{out}}$ now adopts the hue of $\mathbf{m}_0$ and the luma or saturation of $\mathbf{m}_{\text{clip}}$ (depending on the definition used for V_0 and V_{clip}), and because $\mathbf{m}_0 - V_0$ is multiplied by the same g across all channels, their relative proportions remain unchanged, preserving the hue of the original color.

Preserving the luma of the clipped RGB during tone mapping is important for maintaining the contrast of the clipped image, especially in highlight areas. Traditional gamut mapping algorithms often sacrifice or compress the luma component to retain more saturation, which can lead to poor contrast in detailed highlights. The proposed method addresses this by allowing the prioritization of either luma or saturation retention, depending on the specific needs of the image. When luma is prioritized, the approach ensures consistency with the luma of the clipped image but may introduce some desaturation. Conversely, prioritizing saturation may lead to a decrease in luma, particularly in areas of high brightness. The algorithm also provides the flexibility to blend between these two formulations based on the content, allowing for more saturation in flat, saturated areas and more luma in textured highlight areas. This adaptability ensures that the final image achieves a balance between color fidelity and visual contrast, even in challenging scenarios.

Gamut Mapping in YUV Color Space

The proposed gamut mapping algorithm can be directly applied to the YUV color space, which is widely used in digital video and photography. The algorithm follows the same principles as in RGB space, focusing on hue preservation, luma consistency, and saturation management.

Hue h_0 in YUV space is defined using the original chroma components U_0 and V_0 derived from the input values $\mathbf{m}_0$:

$$h_0 = \frac{V_0}{U_0}, \tag{12}$$

where V_0 and U_0 are the red-difference and blue-difference chroma components, respectively. The following algorithm ensures that the gamut mapped output adopt the original hue h_0 after gamut mapping.

Luma Y in YUV space is handled separately from the chroma components, allowing for straightforward luma consistency management. The algorithm adjusts the final output luma Y_{out} to match the clipped luma Y_{clip}, preserving visual contrast and detail while maintaining hue integrity.

To manage saturation and ensure colors remain within the displayable gamut, the algorithm computes a gain factor based on the maximum values of the chroma components before and after clipping:

$$g = \frac{\max(U_{\text{clip}}, V_{\text{clip}})}{\max(U_0, V_0)}, \tag{13}$$

where U_0 and V_0 are the original chroma values before tone mapping, and U_{clip} and V_{clip} are the chroma values after tone mapping and clipping the RGB values, i.e., $\mathbf{m}_{\text{clip}}$ transformed to YUV.

The final output in YUV space is then calculated as:

$$\mathbf{q}_{\text{out}} = [Y_{\text{clip}}, U_0 \times g, V_0 \times g]. \tag{14}$$

Setting $w = 1$ in Eqs.,(8) and (9) renders this YUV-space gamut mapping process equivalent to the RGB version in the previous section. Transforming the YUV values to RGB using a 3x3 conversion matrix, applying the gamut mapping in RGB, and then converting the results back to YUV using the inverse matrix would yield the same outcomes as described in this section. However, working directly in YUV simplifies the process when the input is already in YUV format by leveraging the natural separation of luma and chroma.

Experimental Results

We validated the proposed gamut mapping algorithm by assessing its ability to preserve hue lines in CIELAB color space and visualize constant hues through gradient bars. The experiments were designed to evaluate the following key aspects:

1. **Hue Preservation:** The algorithm's capability to maintain hue consistency across varying saturation and luminance levels.

2. **Luminance Consistency:** We examined how well the algorithm maintains the luminance of pixels before and after gamut mapping, with particular emphasis on highlight areas.

3. **Saturation Reduction:** We tested the algorithm's ability to reduce saturation in a controlled manner, especially for colors exceeding the displayable range.

In the first experiment, we evaluated the coherence of hue lines from the Munsell color system under a D65 illuminant at different stages: original hue lines without tone mapping, hue lines after tone mapping with a power function ($Y_{\text{tm}} = Y_0^{1/10}$) and subsequent clipping, and hue lines after applying the same tone mapping but using the proposed gamut mapping algorithm instead of clipping.

We selected a subset of 168 Munsell papers from four primary hue families—10YY, 10RR, 10GG, and 10BB—spanning nine levels of chroma and ten levels of value. Each hue family forms a distinct hue line in CIELAB space. The goal was to observe the shift in hue lines caused by tone mapping and how effectively the proposed gamut mapping algorithm restores coherence.

Figure 1 shows the three stages of this experiment. In the first plot (a), the original hue lines are displayed, showing straight and coherent paths. Plot (b) presents the hue lines after tone mapping and clipping, where distortions and curvature appear due to clipping, particularly affecting chroma. The third plot (c) demonstrates the effectiveness of the proposed gamut mapping, restoring the hue lines to their near-original form by aligning the chroma values more coherently, reducing the hue shifts introduced by clipping.

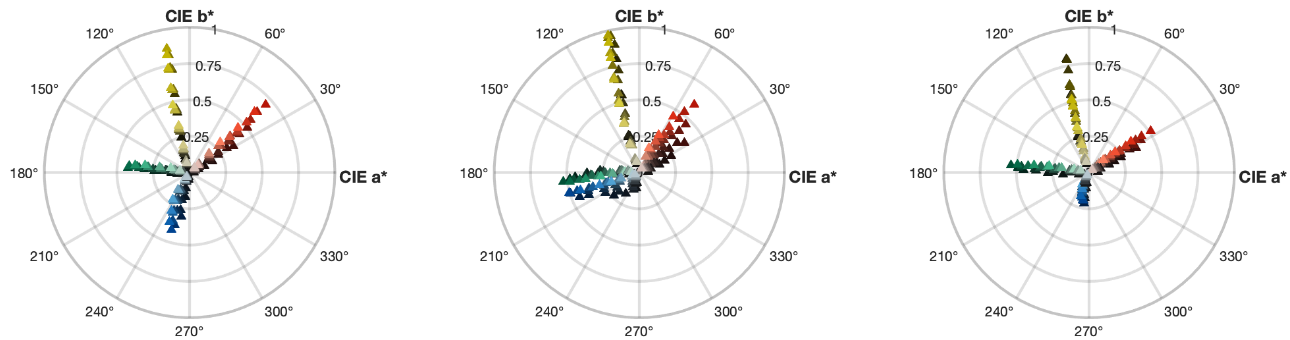
In a second experiment (see fig. 2), we explored the effect of hue retention combined with either luminance retention or saturation retention on color gradients sweeping from black to fully clipped white (1, 1, 1) for two different color examples: skin tone and blue. The results are shown in two vertically stacked plots for each color. The top half of the figure focuses on hue retention with luminance retention, while the bottom half focuses on hue retention with saturation retention.

In the case of *luminance retention* (top half), the goal is to maintain the luminance of the clipped (uncorrected) pixel while preserving the original hue. Both the middle and minimum RGB channels are adjusted to stabilize the hue, ensuring that the overall luminance of the gamut-mapped pixel remains identical to that of the clipped pixel. The dotted lines represent the uncorrected (clipped) RGB values, while the solid lines show the hue-corrected values after gamut mapping. As the intensity approaches clipping, the middle and minimum channels are adjusted together to prevent hue shifts, while preserving the luminance of the clipped pixel.

For *saturation retention* (bottom half), the goal is to preserve the saturation of the clipped pixel while stabilizing the hue to match the original hue before clipping. In this case, only the channel between the maximum and minimum values is adjusted to maintain full chroma while stabilizing the hue. The result is that the color retains its saturation, but the luminance may decrease, particularly in areas of high brightness. Again, the dotted lines indicate the original (clipped) RGB progression, and the solid lines represent the hue-corrected values after gamut mapping.

This experiment highlights the versatility of the proposed gamut mapping algorithm, enabling the user to prioritize either luminance or saturation retention while ensuring that the original hue remains stable as colors approach clipping.

In Figure 3, we illustrate the same concept using 13 color



(a) Original hue lines (b) Tone mapping and clipping (c) Tone mapping and gamut mapping

Figure 1. *Munsell hue lines in CIELAB space for four hue families: 10YY, 10RR, 10GG, and 10BB. (a) Original hue lines, (b) hue lines after tone mapping and clipping, (c) hue lines after tone mapping and applying the proposed gamut mapping algorithm. Radial axis: Chroma (Distance from origin in a^*b^* normalized to 1). Angular axis: Hue direction. The proposed method restores the coherence of the hue lines by aligning the chroma values along a straight path, as opposed to the distorted lines after clipping.*

patches from the standard color checker. The leftmost image shows how gradients are affected when clipped, resulting in noticeable hue shifts. The middle image demonstrates the result of applying both hue and luminance retention, while the rightmost image shows the application of hue and saturation retention. This comparison highlights the trade-offs between clipping, luminance retention, and saturation retention in color correction across different colors. For each patch, the hue deviations (measured in degrees) are annotated within the gradient bars. Although the proposed hue retention algorithm reduces hue shifts in RGB space (where hue deviation is theoretically zero according to eq. 5), small deviations remain in LCH color space. These deviations, however, are significantly smaller in both hue retention methods (middle and right images) compared to the large shifts observed with simple clipping (left image).

In Figure 4, we present the proposed gamut mapping algorithm applied to a real image captured with an iPhone 13 Pro main backfacing camera in ProRaw format.[1] The 2x2 figure shows: (a) the original image with simulated vignetting, (b) the image after vignetting correction and a one f-stop increase in brightness with values clipped above 1, (c) the same image after vignetting correction using the proposed gamut mapping instead of clipping, and (d) the pixel-wise vignetting correction gain map used during correction. The image processed with gamut mapping (c) demonstrates how hue shifts are significantly mitigated (blue vs cyan sky), while more details are preserved, particularly on the boats in the scene. Figure 5 presents a closer look at the sky and sand regions from images (a), (b), and (c), highlighting how the gamut mapping approach further enhances color fidelity, mitigates hue shifts, and preserves subtle textures that are lost when clipping is used.

The experimental results clearly demonstrate the efficacy of the proposed gamut mapping algorithm in preserving hue and maintaining luminance consistency across a variety of scenarios. Whether it is restoring hue coherence in CIELAB space or managing the trade-offs between saturation and luminance retention in gradients, the proposed algorithm consistently outperforms basic clipping methods. The visual results across different experiments validate both the objective benefits, such as reduced hue shifts, and the subjective improvements, like enhanced detail preservation. These findings underline the practical effectiveness of the algorithm for real-time image processing applications, ensuring high color fidelity without introducing artifacts.

[1] This image is from a proprietary dataset and cannot be publicly released in its entirety. We are showing one representative example with permission.

Conclusion

The proposed gamut mapping algorithm provides an efficient and effective way to maintain hue and luminance consistency during tone mapping, with the flexibility to reduce saturation as necessary. This method is computationally lightweight and suitable for real-time applications in digital photography and image processing, where both hardware implementation and color fidelity are critical.

The approach is validated in both RGB and YUV color spaces, showing equivalent performance in preserving color characteristics. Extensive experimental results demonstrate that the algorithm performs reliably across a variety of scenes, including high-dynamic-range images, delivering smooth transitions and maintaining visual coherence. Furthermore, subjective evaluations on images confirm that the algorithm enhances overall image quality, especially in regions prone to color shifts, such as skin tones and saturated colors. This combination of computational efficiency, hue preservation, and adaptability makes the proposed method a valuable tool for applications requiring real-time gamut mapping with high color accuracy.

References

[1] J. Morovič, *Color Gamut Mapping*, John Wiley & Sons, pg. 193-239. (2008).

[2] F. Ebner and M. D. Fairchild, *Development and Testing of a Color Space (IPT) with Improved Hue Uniformity*, Proc. IS&T/SID 6th Color Imaging Conference, pg. 8-13. (1998).

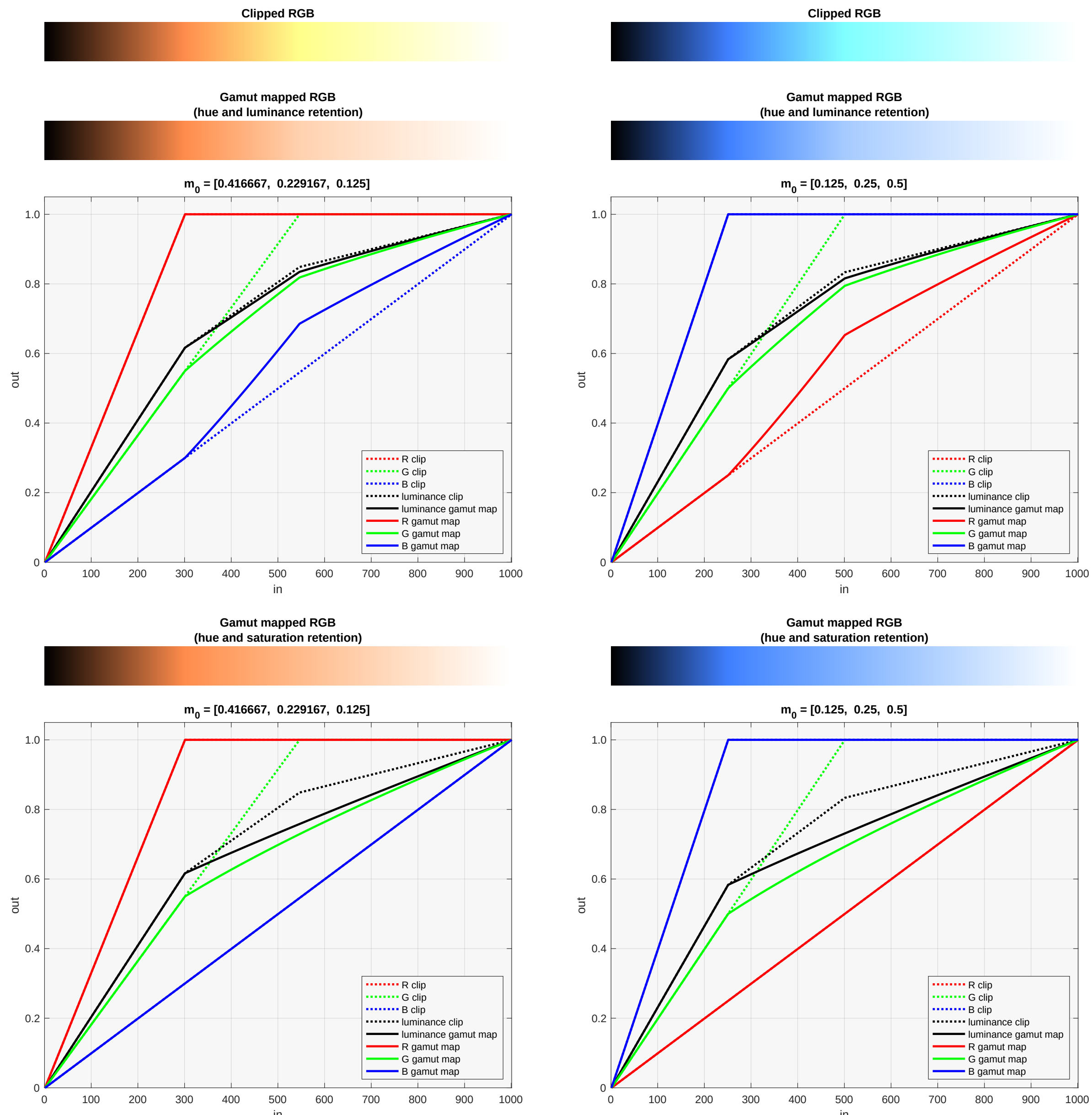


Figure 2. *The figure shows the visual results of hue retention with luminance retention and hue retention with saturation retention for two tones: skin tone (left) and blue tone (right). The top gradient shows the original RGB values as they undergo clipping from black to white. Below that, the second gradient shows the results of gamut mapping with luminance and hue retention, followed by the corresponding plot comparing the clipped and gamut-mapped RGB values. The next gradient demonstrates the results of gamut mapping with saturation and hue retention, followed by its respective plot comparing the clipped and gamut-mapped RGB values.*

[3] Adobe, Digital Negative (DNG) Specification, Version 1.7.1.0, pg. 72. (2023).

[4] Tania Pouli, Alessandro Artusi, Francesco Banterle, Ahmet Oğüz Akyüz, Hans-Peter Seidel, and Erik Reinhard, *Color Correction for Tone Reproduction*, Proc. Color and Imaging Conference, pg. 231-236. (2013).

[5] H. E. Ives, *The Relation Between the Color of the Illuminant and the Color of the Illuminated Object*, Transactions of the Optical Society, Vol. 12, pg. 59-64. (1911).

[6] C. Schlick, Quantization techniques for visualization of high dynamic

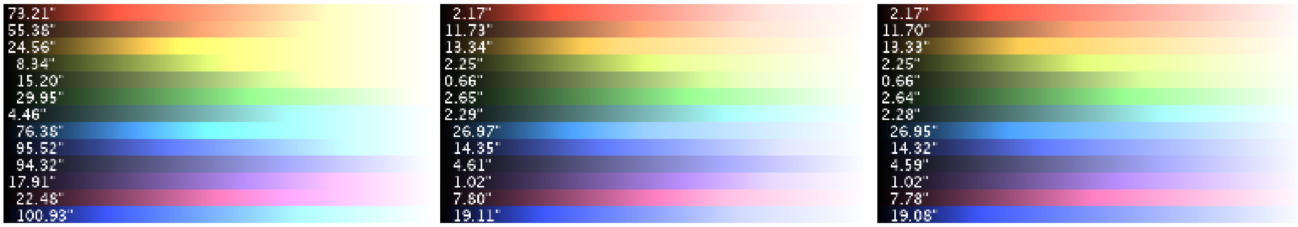


Figure 3. *Gradient sweeps of 13 patches selected from the standard color checker, demonstrating three different methods of color handling: clipped RGB (left), hue retention with luminance retention (middle), and hue retention with saturation retention (right). The numeric labels on each patch indicate the hue deviation (in degrees) between the processed patch (after gamut mapping or clipping) and the original color checker patch, computed as the absolute difference in hue angle (0°–360°) in a CIELAB-based LCH space.*

(a) Image with simulated vignetting

(b) Image with vignetting correction and clipping

(c) Image with vignetting correction and gamut mapping

(d) Vignetting gain correction map

Figure 4. *Results of vignetting correction applied to a iPhone ProRaw image: (a) input image with simulated vignetting, (b) vignetting correction with clipping of excess values, (c) vignetting correction using the proposed gamut mapping instead of clipping, and (d) the pixel-wise vignetting correction gains applied. Gamut mapping effectively mitigates hue shifts (sky, boats, sand).*

range pictures, Proc. 5th Eurographics Workshop on Rendering, pg. 7-18. (1994).

[7] R. Mantiuk et al., Color correction for tone mapping, Computer Graphics Forum, vol. 28, no. 2, pg. 193-202. (2009).

Figure 5. *Zoomed-in comparison of the sky and sand regions from images (a), (b), and (c) in Figure 4. Note the reduced hue shifts in the sky (blue instead of cyan) and the color fidelity of the boats.*