

Chromaticity Gradient Mapping Algorithm Appendix

For reproducibility, we provide additional implementation details for handling edge cases of chromaticity manipulation in our method.

High-Curvature Gradients

Recall that our adjustments shift the chromaticity of each pixel along a level set ℓ of points that share the same distance from ℓ . For ℓ with high curvature, this makes ℓ poorly defined near the center of curvature (see Figure 1). One solution to this problem would be to restrict the maximum curvature of ℓ , but this would prevent plenty of adjustments that are well-posed for a given image. Instead, we use a blend of gradients defined by our curve. Toward the exterior of ℓ 's curvature, we use the gradient defined by ℓ , and as we approach the center of its curvature we use the gradient defined by the *gradient line* connecting ℓ 's endpoints. We linearly blend between these gradients as shown in Figure 1. While this does not guarantee smooth gradients for all gradient curves, it limits remaining discontinuities to the ends of the curve, where users appear to have fewer expectations about what the behavior should be.

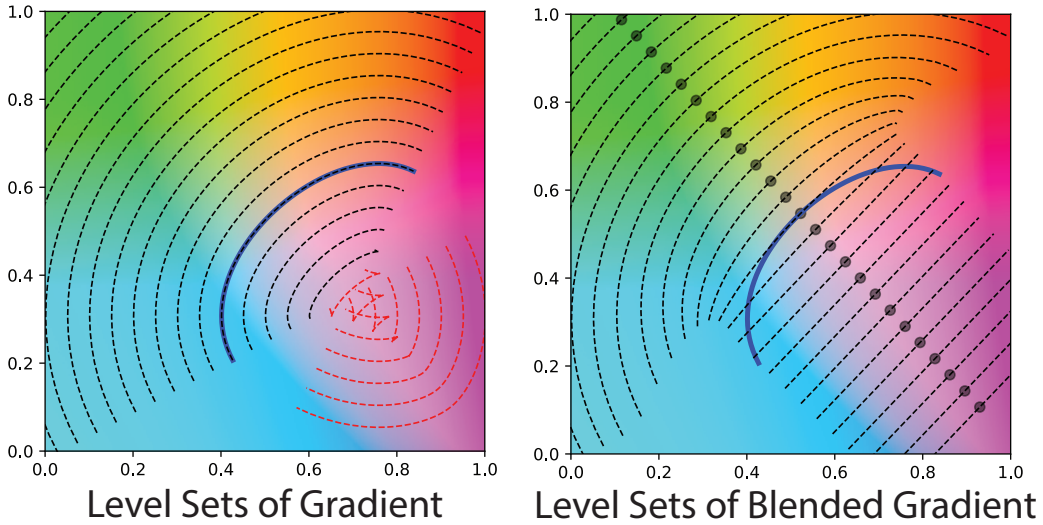


Figure 1: **Level Sets of Gradient Curve:** We show a comparison of level sets computed solely based on the gradient curve (left) versus blending with the gradient line (right). The red dashed curves in the left figure are the ill-defined level sets. The black dots in the right figure are the initial chromaticities for computing the level sets.

RGB Gamut Boundary

The size of the RGB gamut in a^*b^* space is limited and varies with different lightness values. For example, a pixel with lightness 0 or 1 can only take on a single chromaticity, whereas a pixel with lightness of 0.5 can take on a large variety of chromaticities. If a chromaticity adjustment takes a pixel value out of the RGB gamut, then it will have colors outside of the $[0, 1]$ range when converted back to RGB for display. Simply clipping such RGB values will generally change the lightness of the resulting pixel, violating the lightness neutrality of adjustments. To solve this, we take any pixel values that leave the RGB gamut and iteratively clip them and reset their lightness value. This process converges on the boundary of our gamut fast enough to run in real-time on every pixel in a GLSL shader.