Colour Gamut Mapping as a Constrained Variational Problem

Ali Alsam[1] and Ivar Farup[2]

[1] Sør-Trøndelag University College, Trondheim, Norway
[2] Gjøvik University College, Gjøvik, Norway

Abstract. We present a novel, computationally efficient, iterative, spatial gamut mapping algorithm. The proposed algorithm offers a compromise between the colorimetrically optimal gamut clipping and the most successful spatial methods. This is achieved by the iterative nature of the method. At iteration level zero, the result is identical to gamut clipping. The more we iterate the more we approach an optimal, spatial, gamut mapping result. Optimal is defined as a gamut mapping algorithm that preserves the hue of the image colours as well as the spatial ratios at all scales. Our results show that as few as five iterations are sufficient to produce an output that is as good or better than that achieved in previous, computationally more expensive, methods. Being able to improve upon previous results using such low number of iterations allows us to state that the proposed algorithm is $O(N)$, N being the number of pixels. Results based on a challenging small destination gamut supports our claims that it is indeed efficient.

1 Introduction

To accurately define a colour three independent variables need to be fixed. In a given three dimensional colour-space the colour gamut is the volume which encloses all the colour values that can be reproduced by the reproduction device or present in the image. Colour gamut mapping is the problem of representing the colour values of an image in the space of a reproduction device: Typically, a printer or a monitor. Furthermore, in the general case, when an image gamut is larger than the destination gamut some image-information will be lost. We therefore redefine gamut mapping as: The problem of representing the colour values of an image in the space of a reproduction device with minimum information loss.

Unlike single colours, images are represented in a higher dimensional space than three, i.e. knowledge of the exact colour values is not, on its own, sufficient to reproduce an unknown image. In order to fully define an image, the spatial location of each colour pixel needs to be fixed. Based on this, we define two categories of gamut mapping algorithms: In the first, colours are mapped independent of their spatial location [1]. In the second, the mapping is influenced by

A.-B. Salberg, J.Y. Hardeberg, and R. Jenssen (Eds.): SCIA 2009, LNCS 5575, pp. 109–118, 2009.

the location of each colour value [2,3,4,5]. The latter category is referred to as spatial gamut mapping.

Eschbach [6] stated that: Although the accuracy of mapping a single colour is well defined, the reproduction accuracy of images isn't. To elucidate this claim, with which we agree, we consider a single colour that is defined by its hue, saturation and lightness. Assuming that such a colour is outside the target gamut, we can modify its components independently. That is to say, if the colour is lighter or more saturated than what can be achieved inside the reproduction gamut, we shift its lightness and saturation to the nearest feasible values. Further, in most cases it is possible to reproduce colours without shifting their hue.

Taking the spatial location of colours into account presents us with the challenge of defining the spatial components of a colour pixel and incorporating this information into the gamut mapping algorithm. Generally speaking, we need to define rules that would result in mapping two colours with identical hue, saturation and lightness to two different locations depending on their location in the image plane. The main challenge is, thus, defining the spatial location of an image pixel in a manner that results in an improved gamut mapping. By improved we mean that the appearance of the resultant, in gamut, image is visually preferred by a human observer. Further, from a practical point of view, the new definition needs to result in an algorithm that is fast and does not result in image artifacts.

It is well understood that the human visual system is more sensitive to spatial ratios than absolute values [7]. This knowledge is at the heart of all spatial gamut mapping algorithms. A definition of spatial gamut mapping is then: The problem of representing the colour values of an image in the space of a reproduction device while preserving the spatial ratios between different colour pixels. In an image spatial ratios are the difference, given some difference metric, between a pixel and its surround. This can be the difference between one pixel and its adjacent neighbors or pixels far away from it. Thus, we face the problem that: Spatial ratios are defined in different scales and dependent on the chosen difference metric.

McCann suggested to preserve the spatial gradients at all scales while applying gamut mapping [8]. Meyer and Barth [9] suggested to compress the lightness of the image using a low-pass filter in the Fourier domain. As a second step the high-pass image information is added back to the gamut compressed image. Many spatial gamut mapping algorithms have been based upon this basic idea [2,10,11,12,4].

A completely different approach was taken by Nakauchi et al. [13]. They defined gamut mapping as an optimization problem of finding the image that is perceptually closest to the original and has all pixels inside the gamut. The perceptual difference was calculated by applying band-pass filters to Fourier-transformed CIELab images and then weighing them according to the human contrast sensitivity function. Thus, the best gamut mapped image is the image having contrast (according to their definition) as close as possible to the original.

Kimmel et al. [3] presented a variational approach to spatial gamut mapping where it was shown that the gamut mapping problem leads to a quadratic programming formulation, which is guaranteed to have a unique solution if the gamut of the target device is convex.

The algorithm presented in this paper adheres to our, previously, stated definition of spatial gamut mapping in that we aim to preserve the spatial ratios between pixels in the image. We start by calculating the gradients of the original image in CIELab colour space. The image is then gamut mapped by projecting the colour values to the nearest, in gamut, point along hue-constant lines. The difference between the gradient of the gamut mapped image and that of the original is then iteratively minimized with the constraint that the resultant colour is a convex combination of its gamut mapped representation and the center of the destination gamut. Imposing the convexity constraint ensures that the resultant colour is inside the reproduction gamut and has the same hue as the original. Further, if the convexity constraint is removed then the result of the gradient minimization is the original image. The scale at which the gradient is preserved is related to the number of iterations and the extent to which we can fit the original gradients into the destination gamut.

The main contributions of this work are as follows: We first present a mathematically elegant formulation of the gamut mapping problem in colour space. Our formulation can be extended to a higher dimensional space than three. Secondly, our algorithm offers a compromise between the colorimetrically optimal gamut clipping and the most successful spatial methods. This latter aspect is achieved by the iterative nature of the methods. At zero iteration level, the result is identical to gamut clipping. The more we iterate the more we approach McCann's definition of an optimal gamut mapping result. The calculations are performed in the three-dimensional colour space, thus, the goodness of the hue preservation is dependent not upon our formulation but the extent to which the hue lines in the colour space are linear. Finally, our results show that as few as five iterations are sufficient to produce an output that is similar or better than previous methods. Being able to improve upon previous results using such low number of iterations allows us to state that the proposed algorithm is: Fast.

2 Spatial Gamut Mapping: A Mathematical Definition

Let's say we have an original image with pixel values $\mathbf{p}(x, y)$ (bold face to indicate vector) in CIELab or any similarly structured colour space. A gamut clipped image can be obtained by leaving in-gamut colours untouched, and moving out-of-gamut colours along staight lines towards $\mathbf{g}$, the center of the gamut on the L axis until they hit the gamut surface. Let's denote the gamut clipped image $\mathbf{p}_c(x, y)$. From the original image and the gamut clipped one, we can define

$$\alpha_c(x,y) = \frac{||\mathbf{p}_c(x,y) - \mathbf{g}||}{||\mathbf{p}(x,y) - \mathbf{g}||}\,, \tag{1}$$

where $||\cdot||$ denotes the L_2 norm of the colour space. Since $\mathbf{p}_c(x,y) - \mathbf{g}$ is parallel to $\mathbf{p}(x,y) - \mathbf{g}$, this means that the gamut clipped image can be obtained as a linear convex combination of the original image and the gamut clipped one,

$$\mathbf{p}_c(x,y) = \alpha_c(x,y)\mathbf{p}(x,y) + (1 - \alpha_c(x,y))\mathbf{g}. \tag{2}$$

Given that we want to perform the gamut mapping in this direction: This is the least amount of gamut mapping we can do. If we want to impose some more gamut mapping in addition to the clipping, e.g., in order to preserve details, this can be obtained by multiplying $\alpha_c(x,y)$ with some number $\alpha_s(x,y) \in [0,1]$ (s for spatial). With this introduced, the final spatial gamut mapped image can be written as the linear convex combination

$$\mathbf{p}_s(x,y) = \alpha_s(x,y)\alpha_c(x,y)\mathbf{p}(x,y) + (1 - \alpha_s(x,y)\alpha_c(x,y))\mathbf{g}. \tag{3}$$

Now, we assume that the best spatially gamut mapped image is the one having gradients as close as possible to the original image. This means that we want to find

$$\min \int ||\nabla \mathbf{p}_s(x,y) - \nabla \mathbf{p}(x,y)||_F^2 \, dA \quad \text{subject to} \quad \alpha_s(x,y) \in [0,1]. \tag{4}$$

where $||\cdot||_F$ denotes the Frobenius norm on $\mathbb{R}^{3\times 2}$.

In Equation (3), everything exept $\alpha_s(x,y)$ can be determined in advance. Let's therefore rewrite $\mathbf{p}_s(x,y)$ as

$$\mathbf{p}_s(x,y) = \alpha_s(x,y)\alpha_c(x,y)(\mathbf{p}(x,y) - \mathbf{g}) + \mathbf{g} \equiv \alpha_s(x,y)\mathbf{d}(x,y) + \mathbf{g}, \tag{5}$$

where $\mathbf{d}(x,y) = \alpha_c(\mathbf{p}(x,y) - \mathbf{g})$ has been introduced. Then, since $\mathbf{g}$ is constant,

$$\nabla \mathbf{p}_s(x,y) = \nabla(\alpha_s(x,y)\mathbf{d}(x,y)), \tag{6}$$

and the optimisition problem at hand reduces to finding

$$\min \int ||\nabla(\alpha_s(x,y)\mathbf{d}(x,y)) - \nabla \mathbf{p}(x,y)||_F^2 \, dA \quad \text{subject to} \quad \alpha_s(x,y) \in [0,1]. \tag{7}$$

This corresponds to solving the Euler–Lagrange equation:

$$\nabla^2(\alpha_s(x,y)\mathbf{d}(x,y) - \mathbf{p}(x,y)) = 0. \tag{8}$$

Finally, in Figure (1) we present a graphical representation of the spatial gamut problem. $p(x,y)$ is the original colour at image pixel (x,y), this value is clipped to the gamut boundary resulting in a new colour $p_c(x,y)$ which is compressed based on the gradient information to a new value $p_s(x,y)$.

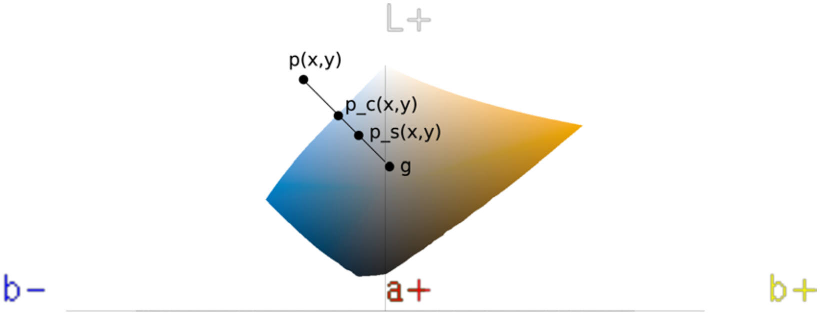

Fig. 1. A representation of the spatial gamut mapping problem. $p(x,y)$ is the original colour at image pixel (x,y), this value is clipped to the gamut boundary resulting in a new colour $p_c(x,y)$ which is compressed based on the gradient information to a new value $p_s(x,y)$.

3 Numerical Implementation

In this section, we present a numerical implementation to solve the minimization problem described in Equation (8) using finite difference. For each image pixel $\mathbf{p}(x,y)$, we calculate forward-facing and backward-facing derivatives. That is: $[\mathbf{p}(x,y)-\mathbf{p}(x+1,y)]$, $[\mathbf{p}(x,y)-\mathbf{p}(x-1,y)]$, $[\mathbf{p}(x,y)-\mathbf{p}(x,y+1)]$, $[\mathbf{p}(x,y)-\mathbf{p}(x,y-1)]$. Based on that, the discrete version of Equation (8) can be expressed as:

$$\begin{aligned} \alpha_s(x,y)\mathbf{d}(x,y) - \mathbf{d}(x+1,y) + \alpha_s(x,y)\mathbf{d}(x,y) - \mathbf{d}(x-1,y) \\ +\alpha_s(x,y)\mathbf{d}(x,y) - \mathbf{d}(x,y+1) + \alpha_s(x,y)\mathbf{d}(x,y) - \mathbf{d}(x,y-1) \\ = \mathbf{p}(x,y) - \mathbf{p}(x+1,y) + \mathbf{p}(x,y) - \mathbf{p}(x-1,y) \\ +\mathbf{p}(x,y) - \mathbf{p}(x,y+1) + \mathbf{p}(x,y) - \mathbf{p}(x,y-1) \end{aligned} \tag{9}$$

where $\alpha_s(x,y)$ is a scalar. Note that in Equation (9) we assume that $\alpha_s(x+1,y)$, $\alpha_s(x-1,y)$, $\alpha_s(x,y+1)$, $\alpha_s(x,y-1)$ are equal to one. This simplifies the calculation, but makes the convergence of the numerical scheme slightly slower.

We rearrange Equation (9) to get:

$$\begin{aligned} \alpha_s(x,y)\mathbf{d}(x,y) \\ = [4 \times \mathbf{p}(x,y) - \mathbf{p}(x+1,y) - \mathbf{p}(x-1,y) \\ -\mathbf{p}(x,y+1) - \mathbf{p}(x,y-1) \\ +\mathbf{d}(x+1,y) + \mathbf{d}(x-1,y) \\ +\mathbf{d}(x,y+1) + \mathbf{d}(x,y-1)] \times \frac{1}{4} \end{aligned} \tag{10}$$

To solve for $\alpha_s(x,y)$, we use least squares. To do that we multiply both sides of the equality by $\mathbf{d}^T(x,y)$ where T denotes vector transpose operator.

$$\begin{aligned}&\alpha_s(x,y)\mathbf{d}^T(x,y)\mathbf{d}(x,y)\\ &= \mathbf{d}^T(x,y)[4 \times \mathbf{p}(x,y) - \mathbf{p}(x+1,y) - \mathbf{p}(x-1,y)\\ &\quad -\mathbf{p}(x,y+1) - \mathbf{p}(x,y-1)\\ &\quad +\mathbf{d}(x+1,y) + \mathbf{d}(x-1,y)\\ &\quad +\mathbf{d}(x,y+1) + \mathbf{d}(x,y-1)] \times \frac{1}{4}\end{aligned} \tag{11}$$

where $\mathbf{d}^T(x,y)\mathbf{d}(x,y)$ is the vector dot product, i.e. a scalar. Finally, to solve for $\alpha_s(x,y)$ we divide both sides of the equality by $\mathbf{d}^T(x,y)\mathbf{d}(x,y)$, i.e.:

$$\begin{aligned}&\alpha_s(x,y)\\ &= \mathbf{d}^T(x,y)[4 \times \mathbf{p}(x,y) - \mathbf{p}(x+1,y) - \mathbf{p}(x-1,y)\\ &\quad -\mathbf{p}(x,y+1) - \mathbf{p}(x,y-1)\\ &\quad +\mathbf{d}(x+1,y) + \mathbf{d}(x-1,y)\\ &\quad +\mathbf{d}(x,y+1) + \mathbf{d}(x,y-1)] \times \frac{1}{4} \times \frac{1}{\mathbf{d}^T(x,y)\mathbf{d}(x,y)}\end{aligned} \tag{12}$$

To insure that $\alpha_s(x,y)$ has values in the range [0 1], we clip values greater than one or less than zero to one, i.e. *if* $\alpha_s(x,y) > 1 \quad \alpha_s(x,y) = 1$ and *if* $\alpha_s(x,y) < 0 \quad \alpha_s(x,y) = 1$, the last one to reset the calculation if the iterative scheme overshoots the gamut compensation.

At each iteration level we update $\mathbf{d}(x,y)$, i.e.:

$$\mathbf{d}(x,y)^{i+1} = \alpha_s(x,y)^i \times \mathbf{d}(x,y)^i \tag{13}$$

The result of the optimization is a map, $\alpha_s(x,y)$, that has values in the range [0 1], where zero takes the value of the clipped pixel $\mathbf{d}(x,y)$ to the average of the gamut and one results in no change.

Clearly, the description given in Equation (12) is an extension of the spatial domain solution of a Poisson equation. It is an extension because we introduce the weights $\alpha_s(x,y)$ with the [0 1] constraint. We solve the optimization problem using Jacobi iteration, with homogenous Neumann boundary conditions to ensure zero derivative at the image boundary.

4 Results

Figures 2 and 3 shows the result when gamut mapping two images. From the α_s maps shown on the right hand side of the figures, the inner workings of the algorithm can be seen. At the first stages, only small details and edges are corrected. Iterating further, the local changes are propagated to larger regions in order to maintain the spatial ratios. Already at two iterations, the result resembles closely those presented in [4], which is, according to Dugay et al. [14] a state-of-the-art algorithm. For many of the images tried, an optimum seems to be found around five iterations. Thus, the algorithm is very fast, the complexity of each iteration being $O(N)$ for an image with N pixels.

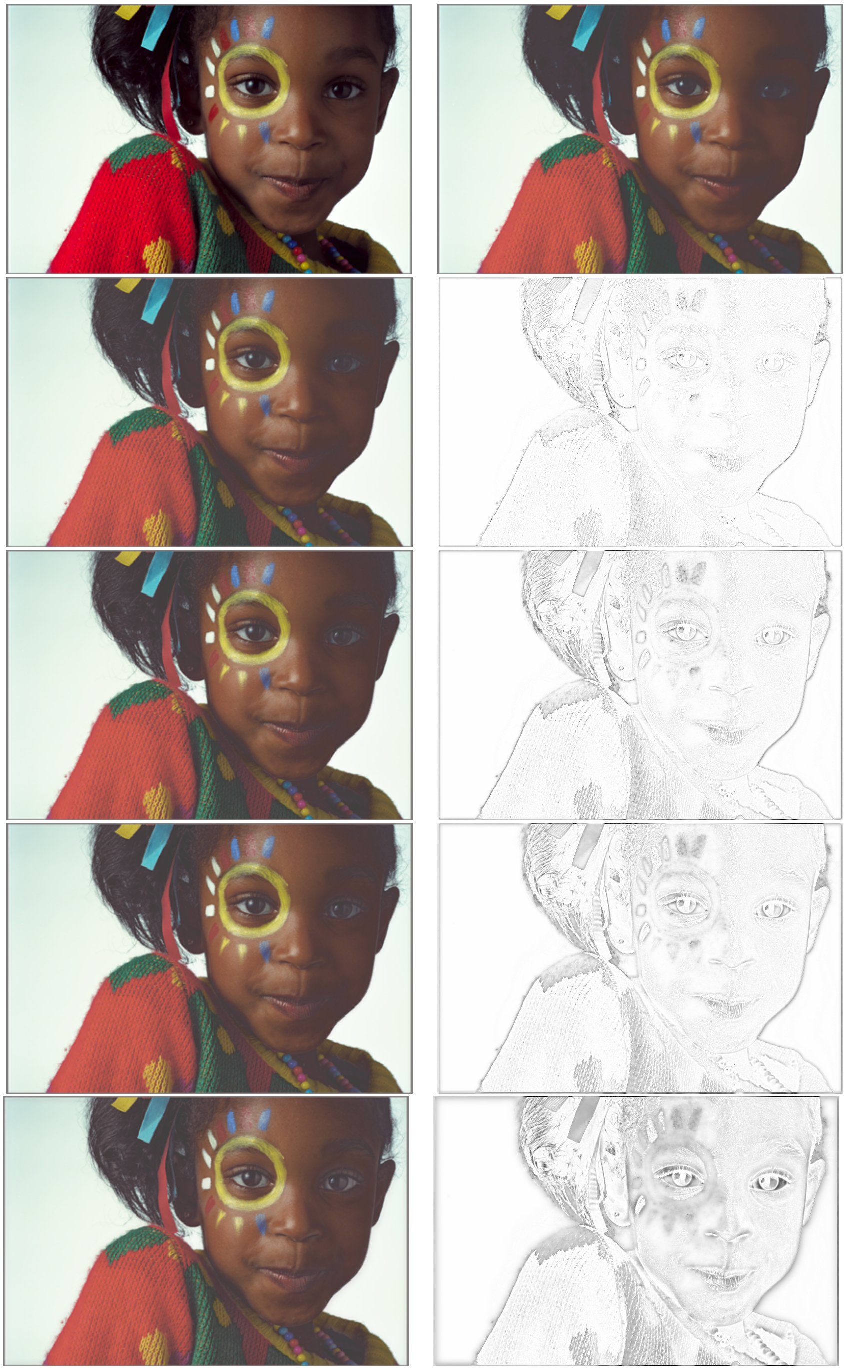

Fig. 2. Original (top left) and gamut clipped (top right) image, resulting image (left column) and α_s (right column) for running the proposed algorithm with 2, 5, 10, and 50 iterations of the algorithm (top to bottom)

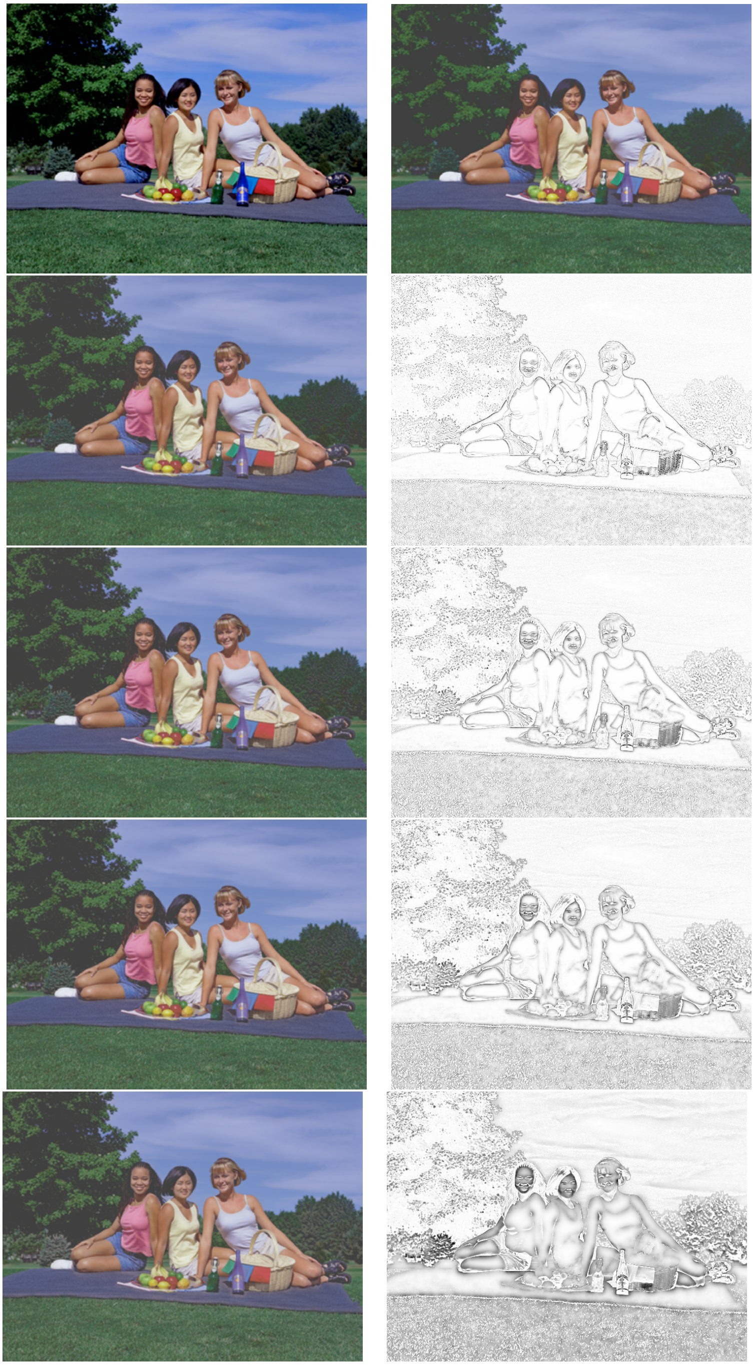

Fig. 3. Original (top left) and gamut clipped (top right) image, resulting image (left column) and α_s (right column) for running the proposed algorithm with 2, 5, 10, and 50 iterations of the algorithm (top to bottom)

As part of this work, we have experimented with 20 images which we mapped to a small destination gamut. Our results shows that keeping the iteration level below twenty results in improved gamut mapping with no visible artifacts. Using a higher number of iterations results in the creation of halos at strong edges and the desaturation of flat regions. A trade-off between these tendencies can be made by keeping the number of iterations below twenty. Further, a larger destination gamut would allow us to recover more lost information without artifacts. We thus recommend that the number of iterations is calculated as a function of the size of the destination gamut.

5 Conclusion

Using a variational approach, we have developed a spatial colour gamut mapping algorithm that performs, at least, as well as state-of-the-art algorithms. The algorithm presented is, however, computationally very efficient and lends itself to implementation as part of an imaging pipeline for commercial applications. Unfortunately, it also shares some of the minor disadvantages of other spatial gamut mapping algorithms: halos and desaturation of flat regions for particularly difficult images. Currently, we working on a modification of the algorithm that incorporates knowledge of the strength of the edge. We believe that this modification will solve or at least reduce strongly these minor problems. This is, however, left as future work.

References

1. Morovič, J., Ronnier Luo, M.: The fundamentals of gamut mapping: A survey. Journal of Imaging Science and Technology 45(3), 283–290 (2001)
2. Bala, R., de Queiroz, R., Eschbach, R., Wu, W.: Gamut mapping to preserve spatial luminance variations. Journal of Imaging Science and Technology 45(5), 436–443 (2001)
3. Kimmel, R., Shaked, D., Elad, M., Sobel, I.: Space-dependent color gamut mapping: A variational approach. IEEE Trans. Image Proc. 14(6), 796–803 (2005)
4. Farup, I., Gatta, C., Rizzi, A.: A multiscale framework for spatial gamut mapping. IEEE Trans. Image Proc. 16(10) (2007), doi:10.1109/TIP.2007.904946
5. Giesen, J., Schubert, E., Simon, K., Zolliker, P.: Image-dependent gamut mapping as optimization problem. IEEE Trans. Image Proc. 6(10), 2401–2410 (2007)
6. Eschbach, R.: Image reproduction: An oxymoron? Colour: Design & Creativity 3(3), 1–6 (2008)
7. Land, E.H., McCann, J.J.: Lightness and retinex theory. Journal of the Optical Society of America 61(1), 1–11 (1971)
8. McCann, J.J.: A spatial colour gamut calculation to optimise colour appearance. In: MacDonald, L.W., Luo, M.R. (eds.) Colour Image Science, pp. 213–233. John Wiley & Sons Ltd., Chichester (2002)
9. Meyer, J., Barth, B.: Color gamut matching for hard copy. SID Digest, 86–89 (1989)
10. Morovič, J., Wang, Y.: A multi-resolution, full-colour spatial gamut mapping algorithm. In: Proceedings of IS&T and SID's 11th Color Imaging Conference: Color Science and Engineering: Systems, Technologies, Applications, Scottsdale, Arizona, pp. 282–287 (2003)

11. Eschbach, R., Bala, R., de Queiroz, R.: Simple spatial processing for color mappings. Journal of Electronic Imaging 13(1), 120–125 (2004)
12. Zolliker, P., Simon, K.: Retaining local image information in gamut mapping algorithms. IEEE Trans. Image Proc. 16(3), 664–672 (2007)
13. Nakauchi, S., Hatanaka, S., Usui, S.: Color gamut mapping based on a perceptual image difference measure. Color Research and Application 24(4), 280–291 (1999)
14. Dugay, F., Farup, I., Hardeberg, J.Y.: Perceptual evaluation of color gamut mapping algorithms. Color Research and Application 33(6), 470–476 (2008)