

Cluster-Based Color Space Optimizations

Cheryl Lau¹

Wolfgang Heidrich¹

Rafał Mantiuk²

¹University of British Columbia

²Bangor University

Abstract

Transformations between different color spaces and gamuts are ubiquitous operations performed on images. Often, these transformations involve information loss, for example when mapping from color to grayscale for printing, from multispectral or multiprimary data to tristimulus spaces, or from one color gamut to another. In all these applications, there exists a straightforward “natural” mapping from the source space to the target space, but the mapping is not bijective, resulting in information loss due to metamerism and similar effects.

We propose a cluster-based approach for optimizing the transformation for individual images in a way that preserves as much of the information as possible from the source space while staying as faithful as possible to the natural mapping. Our approach can be applied to a host of color transformation problems including color to gray, gamut mapping, conversion of multispectral and multiprimary data to tristimulus colors, and image optimization for color deficient viewers.

1. Introduction

Color space transformations are among the most common imaging operations. Whether the goal is to print a color image on a grayscale printer, to fuse multispectral and multi-primary images into a tristimulus image, or to map from one device gamut to another, the task is to find color transformations that respect the limitations of the target space while remaining faithful to the original image content.

Unfortunately, color transformation algorithms have to deal with two sources of information loss. The first is a loss of dimensionality, for example, when mapping from multispectral data to RGB, or from RGB to the space of a color deficient viewer. As a result of this dimensionality reduction, different color vectors in the source space can be mapped to the same color in the target space, an effect that is called *metamerism*. Figure 1 shows two examples of this problem: color to grayscale mapping (top row) and mapping from a six primary space to a three primary RGB

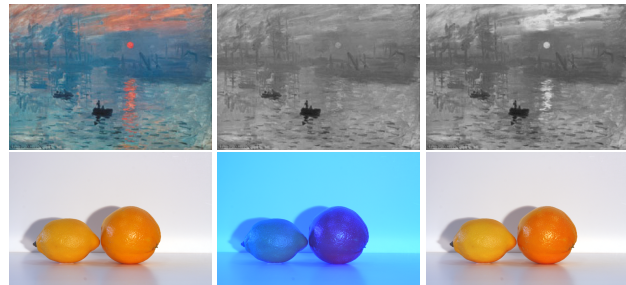


Figure 1. Two examples of color space optimizations. Top row: color to gray conversion. Left: input image; Center: CIE L^* channel; Right: optimized output. Bottom row: conversion of a 6 primary image to RGB. Left: first 3 input channels captured by an RGB camera; Center: remaining 3 input channels captured by the same camera with a blue filter; Right: optimized RGB image preserving the difference between the lemon and the orange. *Original image (top row) from Gooch et al. [9]*

space (bottom row). In both cases, a straightforward, “natural” mapping exists (CIE L^* in the first example, and just using the first 3 primaries of the 6 channel image in the second example). However, in both cases this natural mapping suffers from substantial information loss. Our goal is to define an optimization procedure that preserves the information while also staying close to the natural mapping. Results for our method are shown in the right column.

The second cause of information loss affects situations such as gamut mapping, where the source and target spaces have the same dimensionality but represent different subspaces of that multidimensional space. While it is possible to define a bijective mapping from one color space to another, the goal is again to define a mapping that respects overall image appearance, meaning that source colors that also fall within the target space should not move drastically.

In this paper, we propose a cluster-based algorithm for dealing with both types of information loss in a single, coherent framework. Our method starts with a standard projection from the source to the target space. This standard mapping defines a target appearance for our optimization. We then adopt a semi-local approach that operates on clusters. First, we group pixels into clusters according to both their spatial relationship and their similarity in the source

color space. We then define a graph that describes “conflicts” between clusters, i.e. clusters in the same image region that represent different colors in source space but map to similar colors in target space. Then, we find optimal cluster translations that restore lost contrast while maintaining proximity to the original colors. Finally, we transfer the cluster movements back to the pixels during a blending step.

Since our clustering is based on both spatial and chromatic spacing, we have in effect chosen a semi-local approach over a pure global or pure local solution. This choice is motivated by a number of factors. First, local contrast is more important to human perception than absolute intensities (see, e.g. [17]). Also, operating globally between all parts of the image introduces a heavily constrained system that can become very difficult to satisfy. On the other hand, a completely local method would destroy relationships between similar, non-neighboring colors in a local region, for example patches of sky shining through the branches of a tree. Both issues are avoided by our semi-local clustering. However, we note that the tradeoff between local and global operation can be controlled through the relative weighting in the spatial and chromatic dimensions while performing the clustering steps.

2. Related Work

A variety of work exists on color space optimizations for specific applications as summarized in the following. However, to our knowledge, there is thus far no single method that can handle the full range of applications in a single, coherent fashion.

Color to Gray. Detail preserving mappings from color images to grayscale have been the subject of a lot of research in recent years. The proposals range from pixel-level optimization between pairs of pixels in the image [9] over global optimizations [28, 11, 16, 12], to gradient-based methods [30, 24]. Other work focuses on perceptual accuracy based on the Helmholtz-Kohlrausch effect [29].

In contrast to these approaches, our method focuses on relationships between neighboring areas instead of pixels and only considers spatially close cluster pairs, making our optimization less constrained. As a result, our optimization remains more faithful to the original mapping, while avoiding artifacts such as halos that arise in local filtering methods [29].

Gamut Mapping is a standard problem in imaging, and we refer the interested reader to Morovic’s book [21] for a detailed summary of the state of the art. Gamut mapping algorithms can be categorized as global or local. Global methods map each point in the source gamut to a point in the target gamut. Spatial, or local, methods take into account local neighborhoods around pixels and have the flexibility of a one-to-many mapping. We mention specifically the Hue-angle Preserving Minimum ΔE_{ab} (HPMINDE) map-

ping, which is a standard global method used as a baseline for evaluating gamut mapping algorithms [7]. It clips out-of-gamut points to their closest points on the target gamut boundary within constant hue angle planes. Spatial gamut mapping methods can be further divided into methods that decompose images into bands and successively add detail layers [20, 22], retinex based methods [19], and optimization methods based on image difference metrics [23, 13, 1]. The prevailing issue with spatial gamut mapping algorithms is the tendency to produce halo artifacts. Although mitigated by the use of edge preserving filters [34], results may not be entirely halo free.

Multispectral Image Fusion has the goal of combining multiple spectral channels or primaries into a single fused image that contains more information than any one spectral image alone. Again we refer to a survey article [10] for an overview of the subject. False color band replacement is a popular way of visualizing spectral images which replaces each of the R, G, and B channels with an image from a single spectral band or principal component. Other types of methods include intensity component replacement [5], PCA-based [6], pyramidal [32], wavelet-based [18], region-based [25], and gradient-based methods [26]. Some methods focus on combining visible and infrared (IR) images [3, 33, 14, 31]. Most fusion methods generate a false color image relevant to a given task, but our goal is to generate a natural color image that also shows details from many spectral bands.

Color Deficient Viewers. Dichromats are humans lacking one of the three types of cones found in people with normal color vision. Several researchers have recently addressed the problem of optimizing color images such that color deficient viewers are able to discriminate between different objects they would normally not be able to distinguish. This problem is called *daltonization* [8]. Some methods are extensions of color to gray algorithms [27, 15], while others are based on detailed psychophysical modeling [8].

3. Method

Our cluster-based framework locally improves contrast in five steps. 1) We project the original image to the target space to get the initial mapped image whose contrast we aim to improve. 2) We cluster pixels spectrally and spatially to get areas that may exhibit local contrast. 3) We create a graph connecting spatially close clusters, where graph edges represent the local contrast between neighboring clusters. 4) We solve for new cluster colors in the target space with a least squares optimization that aims to preserve original contrast. 5) Instead of applying the same translation to all pixels in a cluster, we calculate new pixel colors using a weighted blend of cluster translations. To summarize, our cluster-based method groups pixels into clusters according to their spectral and spatial similarities, improves contrast

between clusters by translating the clusters to optimal colors within the target space, and transfers those translations back to the pixels.

3.1. Projection to Target Space

One of our goals is to maintain naturalness, where what is “natural” is defined by some mapping that projects the input image from its source space \mathcal{S} in \mathbb{R}^n to its target space \mathcal{T} in \mathbb{R}^m . Our optimization aims to combat the local contrast loss imposed by this initial mapping, while remaining as faithful to it as possible. For each application, we assume a standard mapping from source space to target space exists. For example, we get an initial gamut mapped image by applying HPMINDE clipping to map the source image to the target gamut. To project color to gray, we use standard mappings such as luma or CIE L^* , but any projection from the source space to the target space could be used to get the initial mapped image. Where possible, we first losslessly transform the source space into a perceptually uniform space such as CIE LUV in order to use L_2 distance as a perceptual metric during clustering and optimization.

3.1.1 Target Space Constraint Representation

For certain applications such as gamut mapping, it is necessary to constrain the mapped colors to a finite area, or gamut \mathcal{T} , within the target space \mathbb{R}^m . Since these constraints usually do not line up with the major axes of the target space, we use a voxel grid to encode the target gamut. This approach enables us to compute signed distance fields from the boundary, which are useful for cluster mapping and in turn enforcing the constraints in the optimization.

For the specific application of gamut mapping, we represent the target gamut with a voxel grid in LCH space, the polar equivalent of CIE LUV. The initial mapping between \mathcal{S} and \mathcal{T} is given by the HPMINDE algorithm, which operates within planes of constant hue that are easily represented in this polar space. For a given color, we retrieve its constant hue plane, shown in Figure 2, and map the color to the closest point on the gamut boundary using L_2 distance. The standard HPMINDE process applies this mapping independently to each pixel in the source image. As a result, many similar colors outside the gamut will be mapped directly onto the boundary, which results in a loss of texture detail. Instead, we propose a new cluster-based mapping that preserves structure within such a group of colors. A cluster with radius r is mapped inside the gamut boundary by mapping the cluster’s mean to the closest point on the distance field’s level set that is r away from the boundary.

3.2. Clustering

While a pure local method operates on pixels, ignoring relationships between areas, and a pure global method op-

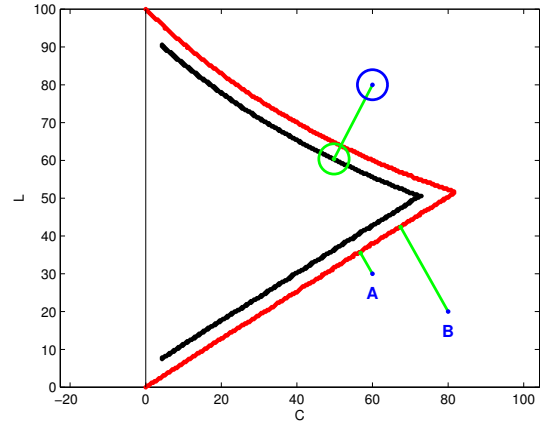


Figure 2. 2D slice through gamut along a plane of constant hue. HPMINDE maps out-of-gamut points (blue) to their closest points (green) on the gamut boundary (red). To map a cluster with radius r , we precompute a signed distance field for the gamut boundary and map the cluster to the level set $-r$ (black). It is possible for the HPMINDE mapping to invert lightness values L for a pair of points (A and B).

erates on colors, ignoring spatial relationships, our semi-local method employs an intermediate approach that works with clusters. To separate the image into areas that may exhibit local contrast between each other, we cluster pixels in spatio-chromatic space \mathcal{U} , in \mathbb{R}^{n+2} , formed by adding 2D spatial dimensions to the source space. Each pixel has a feature vector $\mathbf{u} = (\mathbf{s}, x', y')$ where $\mathbf{s} \in \mathcal{S}$ and (x', y') are the (x, y) pixel coordinates multiplied by a spatial weight that defines the tradeoff between spatial and chromatic dimensions. In practice, we base the spatial weight on the number of JNDs (just noticeable difference units) per length subtended by a 2° visual field and take into account image size, viewing distance, and monitor resolution. We cluster the pixels using k-means++ [2]. Figure 3(b) shows an example output of the clustering step.

In order to preserve and account for intra-cluster detail, we calculate cluster means and covariance matrices, defining ellipsoids, which are used later in the optimization and blending steps. For simplicity, each cluster is also approximated by sphere with radius r equal to the average length of the cluster’s principal components.

3.3. Graph Creation

We represent areas of local contrast by creating a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ connecting spatially close clusters. Graph edges represent local contrast between clusters or, more specifically, the relative original contrast our optimization aims to preserve. Since we focus on enhancing local contrast over global contrast, we only consider contrast between spatially close clusters and exclude spatially distant pairs to avoid

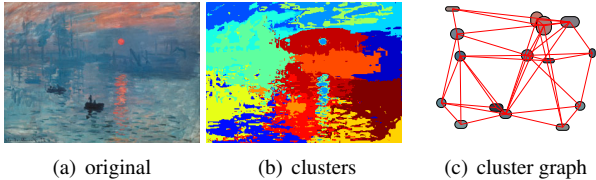


Figure 3. Pixels of the original image are clustered spectrally and spatially into the clusters shown color coded in (b). A graph (c) is created in which each cluster is a vertex. Edges connect spatially close clusters and represent the local contrast we want to preserve.

adding extraneous constraints to the optimization.

Graph edges, based solely upon clusters’ spatial extents (Figure 3(b)), are created by examining the dilated region around a cluster and counting pixels from potential neighbors. For example, the cluster mask in Figure 4(a) is dilated, and the pixels in the dilated region (Figure 4(b)) are counted to produce the histogram of potential neighbors in Figure 4(c). This gives us a histogram per cluster. Clusters i and j are neighbors and (i, j) is an edge in \mathcal{E} if both clusters have large counts above a threshold in each other’s histogram. Figure 3(c) shows the resulting cluster graph. We can optionally consider more spatially distant pairs by adding edges to the two-neighbors or three-neighbors in the original graph.

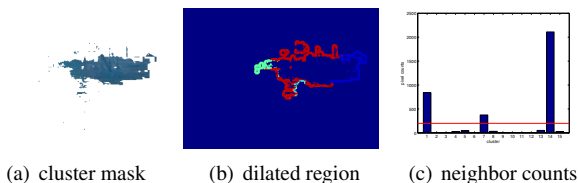


Figure 4. Graph creation. (a) Pixels from a single cluster. (b) Dilated region, color coded by neighbor. (c) Histogram of pixel counts in dilated region. Clusters are neighbors if both clusters have large counts above a threshold (red) in each other’s histogram.

3.4. Optimization

Given the graph, we apply an optimization procedure to translate the clusters within the target space such that contrast is improved between clusters that have lost contrast. At the same time, we would like to remain faithful to the initial projection into the target space. We solve a linear least squares problem for the optimal cluster colors \mathbf{x} that minimize

$$\mathbf{x} = \arg \min_{\mathbf{x}} E_T + wE_M, \quad (1)$$

where E_T is the term that preserves contrast, E_M is a regularization term that stays close to the initial mapping, and w is a parameter weight.

Target Term. The term E_T preserves contrast by matching cluster difference vectors to target vectors \mathbf{t}_{ij} for all edges

(i, j) in \mathcal{E} . The target term is

$$E_T = \sum_{(i,j) \in \mathcal{E}} \tau_{ij} (\|\mathbf{x}_j - \mathbf{x}_i - \mathbf{t}_{ij}\|)^2, \quad (2)$$

where τ_{ij} is a weight on edge (i, j) . To increase the contrast between a cluster pair, we lengthen its initial vector \mathbf{m}_{ij} between the mapped clusters by an amount based on the visible contrast lost. Therefore, we set the target vector \mathbf{t}_{ij} to the initial mapped vector \mathbf{m}_{ij} with magnitude m_{ij} lengthened by some additional magnitude a_{ij} , as defined below.

$$\begin{aligned} \mathbf{t}_{ij} &= \frac{m_{ij} + a_{ij}}{m_{ij}} \mathbf{m}_{ij} \\ a_{ij} &= k \cdot \text{scale}(\psi_{ij} * (o_{ij} - \|f(\mathbf{m}_j) - f(\mathbf{m}_i)\|)) \\ \psi_{ij} &= \frac{1}{1 + e^{-\kappa(\|f(\mathbf{m}_j) - f(\mathbf{m}_i)\| - c)}} \end{aligned} \quad (3)$$

Cluster pairs that lose a lot of contrast, such as those with different source colors that map to the same target color, need their contrast restored the most and thus receive the largest a_{ij} values up to the user defined parameter k . The additional magnitude a_{ij} is based on how far apart clusters are before and after the initial mapping. The L_2 distance o_{ij} between the clusters in the source space is compared to $\|f(\mathbf{m}_j) - f(\mathbf{m}_i)\|$, the comparable L_2 distance from the target space, where the function $f(\mathbf{m})$ transforms the target space clusters \mathbf{m} to make the distances from the target space comparable to the distances in the source space. Since we do not need to enhance the contrast for cluster pairs that are already visibly different, we multiply by a sigmoidal weight ψ_{ij} , giving those cluster pairs less additional magnitude. The sigmoid, determined empirically from our image data, is centered around $c = 15$ JND with steepness $\kappa = 80$. The $\text{scale}()$ function clamps negatives and makes the max equal one, resulting in a_{ij} values in $[0, k]$.

Critical edges are the cluster pairs that most require contrast enhancement. Edge (i, j) is critical if the initial mapping causes clusters i and j to overlap more in the target space than they did in the source space. The per edge weight $\tau_{ij} = e^{-x}$ gives more weight to edges that are closer to critical edges in the graph, allowing these edges to better match their targets. The per cluster weight $\tau_i = 1 - e^{-(x+0.1)}$ in Equation 4 gives less weight to clusters that are closer to critical clusters, allowing these clusters more freedom to move. Here, x is the path length from edge (i, j) or cluster i to the nearest critical edge or cluster in \mathcal{G} .

When the source and the target spaces have the same dimensions, i.e. $\mathbb{R}^n = \mathbb{R}^m$, we set the target vectors to the original source vectors \mathbf{o}_{ij} between clusters in lieu of Equation 3, preserving the original vector directions and magnitudes.

Regularization Term. The term E_M keeps the output cluster colors close to the initial mapping \mathbf{m} to the target space, thereby maintaining naturalness.

$$E_M = \sum_{i \in \mathcal{V}} \tau_i (\mathbf{x}_i - \mathbf{m}_i)^2 \quad (4)$$

Hue Term. For some applications, such as gamut mapping, where it is important to preserve hue, we add a term E_H to Equation 1 weighted by parameter β . This prevents hue shift by penalizing cluster movement in the direction orthogonal to each cluster’s plane of constant hue. The matrix H is a block diagonal matrix composed of blocks H_1, \dots, H_N where N is the number of clusters. Each block H_i is an $m \times m$ matrix that projects the optimized color \mathbf{x}_i in \mathbb{R}^m onto the space orthogonal to cluster i ’s constant hue plane.

$$E_H = \|H\mathbf{x}\|^2 \quad (5)$$

Achromaticity Term. Where applicable, we preserve achromaticity, ensuring that neutral source colors remain neutral in the target space by adding a term E_L to Equation 1 weighted by parameter γ . For achromatic source colors, the term penalizes cluster movement off the neutral axis of the target space. In the block diagonal matrix Q , structured similar to H , each block Q_i projects the cluster movement vector $\mathbf{x}_i - \mathbf{m}_i$ onto the subspace orthogonal to the neutral axis. In the matrix $L = WQ$, these projections are weighted by the achromaticity weights in the diagonal matrix W . For each cluster, the achromaticity weight is a gaussian function of the original cluster’s chroma value with $\sigma = 10$ JND units.

$$E_L = \|L(\mathbf{x} - \mathbf{m})\|^2 \quad (6)$$

3.4.1 Constrained Optimization

For applications such as gamut mapping, we need to constrain the optimized cluster colors to lie within a finite gamut \mathcal{T} in \mathbb{R}^m . In this case, we perform a constrained optimization in which we solve the linear least squares problem for the optimal cluster colors in \mathbb{R}^m , project the optimized results to the target gamut \mathcal{T} , and iterate. On iteration z , we solve for optimal colors \mathbf{x}^z and project them to \mathcal{T} using projection operator $P(\cdot)$, yielding projected output colors $\mathbf{p}^z = P(\mathbf{x}^z)$.

We add the term E_P to Equation 1 weighted by parameter α to keep the optimized cluster colors \mathbf{x} close to the projected output \mathbf{p} of the previous iteration.

$$E_P = \sum_{i \in \mathcal{V}} (\mathbf{x}_i^z - \mathbf{p}_i^{z-1})^2. \quad (7)$$

3.4.2 Parameter Settings

The parameters in our optimization are set to constants with the exception of k and possibly w . We found that $\alpha = 0.5$, $\beta = 4$, $\gamma = 10$ works well, preventing hue shifts and unwanted chromaticity gains. Setting $w = 0.8$ for the regularization term works well to preserve the realistic nature of the scene, but we can decrease w if we want to allow the optimization to deviate further from the natural mapping. The parameter k , typically set in $[0.1, 1]$, controls the amount of contrast enhancement. This is the only true variable in our approach as it controls the application-specific tradeoff between staying close to the generic mapping and recovering the lost contrast from the source space.

3.5 Blending

Every pixel in a cluster is not exactly represented by its cluster’s mean. In fact, a pixel given by its vector $\mathbf{u} \in \mathcal{U}$ may be located between the means of neighboring clusters in \mathbb{R}^{n+2} , and for these pixels it is better to blend between the results of the neighboring clusters. The optimization step solves for the output cluster colors \mathbf{x} in the target space (or for the projected output \mathbf{p} in the constrained optimization case). Using the mapped cluster colors \mathbf{m} we find difference vectors $\mathbf{d}_i = \mathbf{x}_i - \mathbf{m}_i$ for each cluster i . Instead of applying the same difference vector to every pixel in the cluster, a difference vector \mathbf{d}_q for each pixel q is calculated as a weighted combination of difference vectors over all clusters. For each pixel, the weight ω_{qi} on each cluster’s difference vector is inversely proportional to the pixel’s squared Mahalanobis distance to cluster i . The squared Mahalanobis distance is \mathbf{u} evaluated in the cluster’s ellipsoid equation defined by cluster mean μ and inverse covariance matrix M .

$$\mathbf{d}_q = \sum_{i \in \mathcal{V}} \omega_{qi} \mathbf{d}_i, \quad \text{with} \quad \omega_{qi} = \frac{1}{(\mathbf{u} - \mu_i)^T M_i \mathbf{u}} \quad (8)$$

The pixel is updated by adding its difference vector \mathbf{d}_q to its mapped color vector. In applications, such as gamut mapping, where the source and target spaces are both in \mathbb{R}^n , we can use source clusters to compute difference vectors and update source pixel values, preserving intra-cluster detail. By blending the optimization results, we smooth the output where necessary and avoid quantization-like discontinuities. Artifacts from misclassifying pixels in the clustering step are also subdued since those pixels will get higher weights for the clusters to which they are actually closer.

Finally, to ensure that all colors are within \mathcal{T} , the output pixel colors are clipped to the target space.

4. Results

Our method is able to preserve details that are lost during a standard projection to the target space while remaining close to that projection, which is important for maintaining

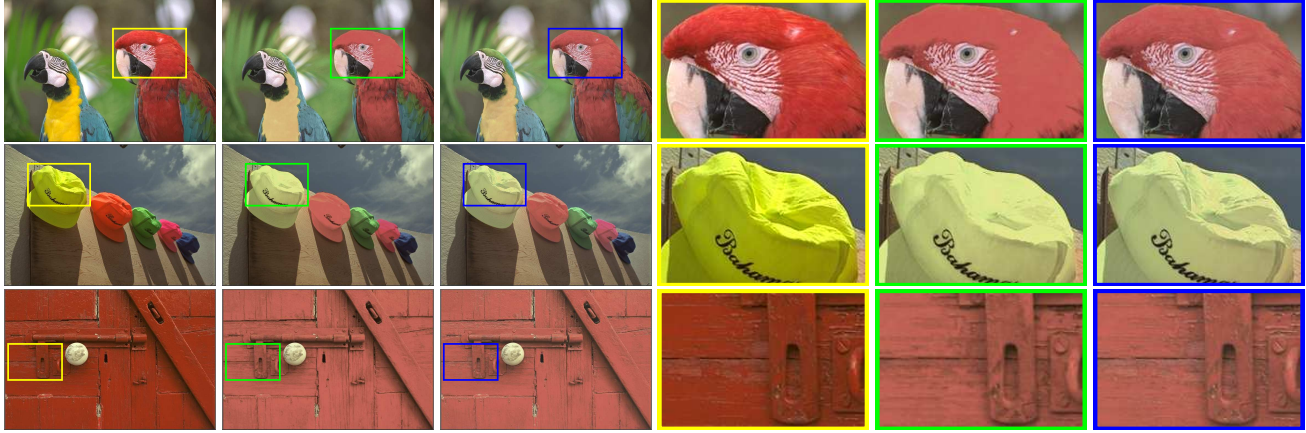


Figure 7. Our framework applied to gamut mapping. Column 1: original RGB images. Column 2: HPMINDE clipping. Column 3: our output. The last three columns are cropped regions of the first three columns showing the details preserved in the feathers and in the wrinkles of the hat. In the close-up of the door, lightness inversions from the initial HPMINDE mapping are fixed in our output. *Original images courtesy of Kodak.*

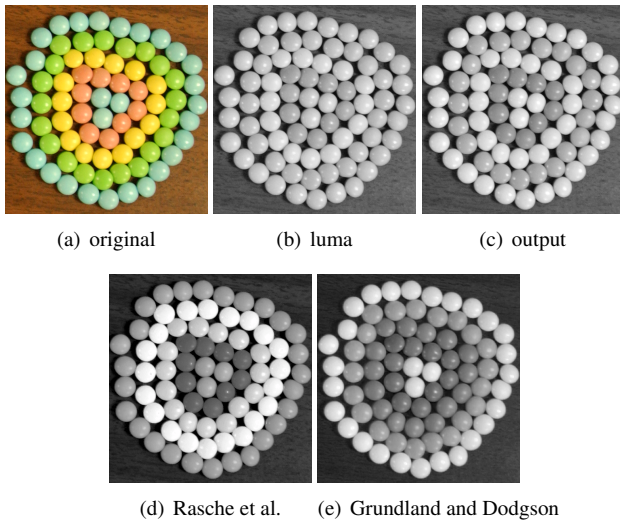


Figure 5. During the standard projection to luma, the rings of candy lose contrast. Our method enhances contrast between the rings without deviating too far from the luma image, producing a more realistic grayscale version than global mappings like those of Rasche et al. [28] and Grundland and Dodgson [11]. *Original image courtesy of Rasche et al. [28]*

naturalness. For example, in Figure 1 (top row) our method restores the contrast between the red and blue elements of the impressionist painting that was lost during the standard L^* conversion to grayscale. The rings of candy in Figure 5 lose most of their contrast during the standard luma conversion. Our method enhances the differences between the rings without moving too far from the initial luma values. This results in a more realistic grayscale version than global mappings such as those from Rasche et al. [28] and Grundland et al. [11]. Figure 6 shows how our method avoids halo

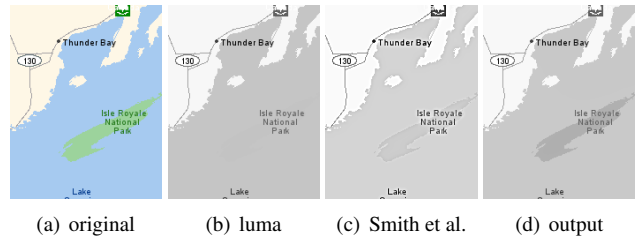


Figure 6. The green island disappears when this image is converted to luma. Our method and the method of Smith et al. [29] both restore contrast, but our method does not contain halo artifacts around the island that are common to methods with local filtering. *Original image courtesy of Yahoo! Maps/NAVTEQ/DigitalGlobe.*

artifacts that arise in methods with local filtering operations such as Smith et al. [29]. Both our method and the method of Smith et al. restore the island visibility, but Smith et al.'s method introduces a halo around the island due to its local unsharp masking step. Through our semi-local, cluster-based optimization approach we can preserve local contrast, stay close to an initial mapping, and avoid halo artifacts.

Figure 7 shows the results of our framework applied to gamut mapping. To illustrate our method, we map to a toy target gamut with less saturated primaries than sRGB. Our results contain details in the out-of-gamut regions that were clipped by HPMINDE. For example, details in the birds' feathers (first row) and wrinkles in the hats (second row) are more visible in our result than in the HPMINDE mapping. As illustrated in Figure 2, it is possible for the HPMINDE mapping to invert the lightness of two points. Such a reversal happens for the image of the door (third row). Our method is able to fix this lightness inversion since our optimization matches targets that are in the original vectors' directions.

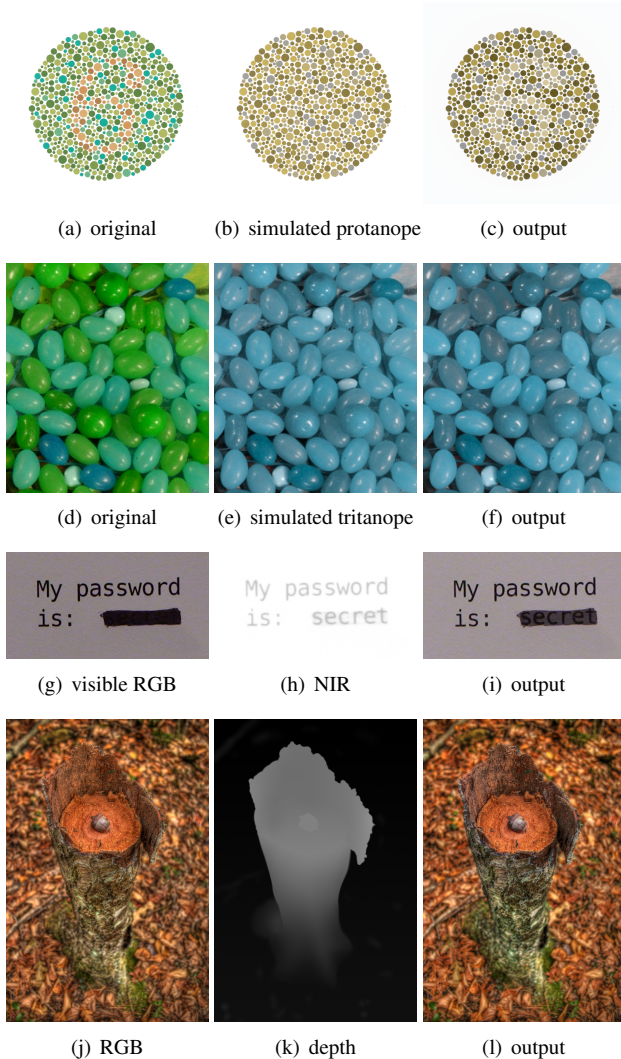


Figure 8. Our framework applied to multichannel image fusion. Rows 1 and 2: Image optimization for color deficient viewers. Our output preserves contrasts that color deficient viewers normally would not see. *Original images courtesy of Wikipedia (row 1) and Rasche et al. [28] (row 2)*. Row 3: Fusion of RGB image and near-infrared image. Our output enhances visibility of the hidden password. Row 4: Fusion of RGB image and depth image (*Image pair courtesy of Justin Manteuffel*). Our output shows improved contrast between the tree stump and the background.

Figure 8 shows the results of our framework applied to converting multispectral or multichannel images to lower dimensional images. In the first two rows, we simulate two types of color deficient vision using Brettel et al. [4]. Protanopes have trouble seeing the digit '6' in the Ishihara image while tritanopes have trouble distinguishing between the green and blue jelly beans. With our method, the '6' emerges from the dot pattern, and the two kinds of jelly beans are different shades of blue. In the third row, we combine an RGB image with a near infrared image. Near

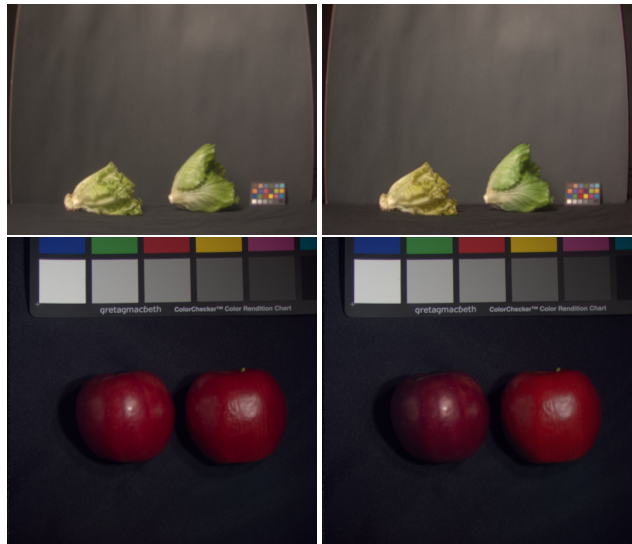


Figure 9. Our framework applied to multispectral images of metameric artificial and real food. Column 1: visible RGB image (standard spectral to RGB conversion). Column 2: Our output increases contrast between the metamers. *Original image (row 2) courtesy of the Columbia Multispectral Image Database.*

infrared light penetrates some inks, as it does in this example to reveal the password that was hidden under black marker. We are able to capture the hidden password in our output. When combining an RGB image with a depth map, the contrast between the tree stump and background is enhanced, and the colors of the tree stump change to show the gradation in the depth map. Figure 1 (second row) contains a multiprimary image of a plastic lemon and real orange. The lemon and orange have similar RGB colors but different spectral responses as evident in the blue filtered image. The six primary image is reduced to an RGB image where the fruits are different colors from each other but still similar enough to their natural colors. Similarly, in Figure 9, the real and artificial foods have different spectral responses, but this is not easily seen by the human visual system or preserved when spectral data is converted to RGB. Our method is able to distinguish between these spectral metamers.

Our method preserves contrast from the source space within the smaller target space while remaining close to a standard projection to the target space to maintain naturalness.

5. Conclusion

In conclusion, we propose a unified framework to solve a wide variety of color space transformation problems using the same method. We apply our method to the specific problems of color to gray conversion, color gamut mapping, image optimization for color deficient viewers, and multispectral image fusion. Our method preserves local contrast

while maintaining a realistic appearance by solving an optimization that matches target differences and stays close to a standard projection to the target space. Our method is semi-local because it operates on clusters, modifying local contrasts between clusters.

6. Acknowledgements

This work was performed under the Dolby Research Chair in Computer Science at the University of British Columbia.

References

- [1] A. Alsam and I. Farup. Colour gamut mapping as a constrained variational problem. *16th Scandinavian Conf. Image Analysis*, 5575:109–118, 2009.
- [2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proc. 18th ACM-SIAM Symp. Discrete Algorithms*, pages 1027–1035, 2007.
- [3] E. Bennett, J. Mason, and L. McMillan. Multispectral bilateral video fusion. *IEEE Trans. Image Process.*, 16(5):1185–1194, 2007.
- [4] H. Brettel, F. Viénot, and J. Mollon. Computerized simulation of color appearance for dichromats. *J. Opt. Soc. Am. A*, 14(10):2647–2655, 1997.
- [5] W. Carper, T. Lillesand, and R. Kiefer. The use of intensity-hue-saturation transformations for merging spot panchromatic and multispectral image data. *Photogramm. Eng. Remote Sens.*, 56:459–467, 1990.
- [6] P. Chavez and A. Kwarteng. Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis. *Photogramm. Eng. Remote Sens.*, 55:339–348, 1989.
- [7] CIE. Guidelines for the evaluation of gamut mapping algorithms. Technical Report CIE 156:2004, 2004.
- [8] Daltonize. <http://www.daltonize.org>, 2011.
- [9] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch. Color2gray: Saliency-preserving color removal. *ACM Trans. Graph.*, 24(3):634–639, 2005.
- [10] A. Goshtasby and S. Nikolov. Image fusion: Advances in the state of the art. *Information Fusion*, 8(2):114–118, 2007.
- [11] M. Grundland and N. A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recogn.*, 40(11):2891–2896, 2007. <http://www.Eyemaginary.com/Portfolio/Publications.html>.
- [12] Y. Kim, C. Jang, J. Demouth, and S. Lee. Robust color-to-gray via nonlinear global mapping. *ACM Trans. Graph.*, 28(5):161, 2009.
- [13] R. Kimmel, D. Shaked, M. Elad, and I. Sobel. Space-dependent color gamut mapping: A variational approach. *IEEE Trans. Image Process.*, 14(6):796–803, 2005.
- [14] D. Krishnan and R. Fergus. Dark flash photography. *ACM Trans. Graph.*, 28(3):96, 2009.
- [15] G. Kuhn, M. Oliveira, and L. Fernandes. An efficient naturalness-preserving image-recoloring method for dichromats. *IEEE Trans. Vis. Comput. Graphics*, 14(6):1747–1754, 2008.
- [16] G. Kuhn, M. Oliveira, and L. Fernandes. An improved contrast enhancing approach for color-to-grayscale mappings. *Vis. Comput.*, 24:505–514, 2008.
- [17] E. H. Land and J. J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61:1–11, 1971.
- [18] H. Li, B. Manjunath, and S. Mitra. Multisensor image fusion using the wavelet transform. *Graph. Models Image Process.*, 57(3):235–245, 1995.
- [19] J. McCann. Lessons learned from mondrians applied to real images and color gamuts. In *Proc. IS&T/SID 7th Color Imaging Conf.*, pages 1–8, 1999.
- [20] J. Meyer and B. Barth. Color gamut matching for hard copy. In *Proc. SID Dig.*, pages 86–89, 1989.
- [21] J. Morovic. *Color Gamut Mapping*. John Wiley & Sons Ltd., 2008.
- [22] J. Morovic and Y. Wang. A multi-resolution, full-colour spatial gamut mapping algorithm. In *Proc. IS&T/SID 11th Color Imaging Conf.*, pages 282–287, 2003.
- [23] S. Nakauchi, S. Hatanaka, and S. Usui. Color gamut mapping based on a perceptual image difference measure. *Color Research Application*, 24(4):280–291, 1999.
- [24] L. Neumann, M. Cadik, and A. Nemcsics. An efficient perception-based adaptive color to gray transformation. In *Proc. Computational Aesthetics*, pages 73–80, 2007.
- [25] G. Piella. A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion*, 4(4):259–280, 2003.
- [26] G. Piella. Image fusion for enhanced visualization: A variational approach. *Int. J. Comput. Vision*, 83(1):1–11, 2009.
- [27] K. Rasche, R. Geist, and J. Westall. Detail preserving reproduction of color images for monochromats and dichromats. *IEEE Comput. Graph. Appl.*, 25(3):22–30, 2005.
- [28] K. Rasche, R. Geist, and J. Westall. Re-coloring images for gamuts of lower dimension. *Comput. Graph. Forum*, 24(3):423–432, 2005.
- [29] K. Smith, P.-E. Landes, J. Thollot, and K. Myszkowski. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Comput. Graph. Forum*, 27(2):193–200, 2008.
- [30] D. A. Socolinsky and L. B. Wolff. Multispectral image visualization through first-order fusion. *IEEE Trans. Image Process.*, 11(8):923–931, 2002.
- [31] S. Süsstrunk and C. Fredembach. Enhancing the visible with the invisible: Exploiting near-infrared to advance computational photography and computer vision. *SID Int. Symp. Dig.*, 2010.
- [32] A. Toet, J. van Ruyven, and J. Valette. Merging thermal and visual images by contrast pyramids. *Opt. Eng.*, 28(7):789–792, 1989.
- [33] X. Zhang, T. Sim, and X. Miao. Enhancing photographs with near infra-red images. In *IEEE Conf. Comput. Vision Pattern Recogn.*, pages 1–8, 2008.
- [34] P. Zolliker and K. Simon. Retaining local image information in gamut mapping algorithms. *IEEE Trans. Image Process.*, 16(3):664–672, 2007.