Laplacian colormaps: a framework for structure-preserving color transformations

D. Eynard, A. Kovnatsky, and M. M. Bronstein
Institute of Computational Science, Faculty of Informatics, University of Lugano, Switzerland

Abstract
Mappings between color spaces are ubiquitous in image processing problems such as gamut mapping, decolorization, and image optimization for color-blind people. Simple color transformations often result in information loss and ambiguities, and one wishes to find an image-specific transformation that would preserve as much as possible the structure of the original image in the target color space. In this paper, we propose Laplacian colormaps, a generic framework for structure-preserving color transformations between images. We use the image Laplacian to capture the structural information, and show that if the color transformation between two images preserves the structure, the respective Laplacians have similar eigenvectors, or in other words, are approximately jointly diagonalizable. Employing the relation between joint diagonalizability and commutativity of matrices, we use Laplacians commutativity as a criterion of color mapping quality and minimize it w.r.t. the parameters of a color transformation to achieve optimal structure preservation. We show numerous applications of our approach, including color-to-gray conversion, gamut mapping, multispectral image fusion, and image optimization for color deficient viewers.

Categories and Subject Descriptors (according to ACM CCS):

1. Introduction
A wide class of image processing problems relies on transformations between color spaces. Some notable examples include gamut mapping, image optimization for color-deficient viewers, and multispectral image fusion. Often, these transformations imply a reduction in the dimensionality of the original color space, resulting in information loss and ambiguities.

Decolorization or color-to-gray conversion is a classical example one frequently encounters when printing a color image on a black-and-white printer. The ambiguity of such a conversion (called metamerism, when many different RGB colors are mapped to the same gray level) may result in the loss of important structures in the image (see Figure 1). Preserving salient characteristics of the original image is thus crucial for a quality color transformation process. These characteristics can be represented in different ways, e.g. as contrasts between color pixels in terms of their luminance and chrominance [G*05], color distances [GD07], image gradients [ZF12] and Laplacians [BD13].

Color-to-gray maps can be classified into global (using the same map for each pixel) and local (or spatial, allowing different pixels with the same color to be mapped to
different gray values, at the advantage of a better perception of color contrasts). Members of the first group include the pixel-based approaches by Gooch et al. [G*05] and Grundland et al. [GD07], and the color-based ones by Rasche et al. [RGW05b], Kuhn et al. [KO*08a], Kim et al. [KJDL09], and Lu et al. [LXJ12]. Among local methods [NvN07, KAC10, ZF12], several aim to preserve information in the gradient domain. Smith et al. [S*08] present a hybrid (local-global) approach that relies on both an image-independent global mapping and a multiscale local contrast enhancement. Lau et al. [LHM11] propose an approach defined as ‘semi-local’, as it clusters pixels based on both their spatial and chromatic similarities. The color mapping problem is solved with an optimization aimed at finding optimal cluster colors such that the contrast between clusters is preserved.

**Gamut mapping** is the process of adjusting the colors of an input image into the constrained color gamut of a given device. Gamut mapping algorithms can be mainly divided into clipping and compression approaches [Mor08]. The former ones change the source colors that fall outside of the destination gamut (e.g. HPMINDE [CIE04, B*06]); the latter also modify the in-gamut colors. Similarly to color-to-gray conversion, gamut mapping methods can also be categorized as global and local. To address metamerism in gamut mapping, local approaches [BdQE10, NHE10, K*05] allow two spatially-distant pixels of equal color to be mapped to different in-gamut colors. Global approaches, conversely, will always apply the same map to two pixels of the same color, regardless of their location. Many gamut mapping algorithms optimize some image difference criterion [NHE10, K*05, AF09, LHM11].

**Color-blind viewers** cannot perceive differences between some given colors, due to the lack of one or more types of cone cells in their eyes [dal, MG88]. Image perception by a color-deficient observer is typically simulated by first applying a linear transformation from a standard color space such as RGB [K*12, BVM07, VBM99], XYZ [MG88, RGW05a], or CIE Lab* [KO*08a, H*07] to a special LMS space, which specifies colors in terms of the relative excitations of the cones. Then, the color domain is reduced in accordance with the color deficiency (typically, by means of a linear transformation in the LMS space [VBM99, K*12, H*07]). Finally, the reduced LMS space is mapped back to RGB.

When trying to adapt an image for a color-blind viewer, one has to ensure that the structure of the original image is not lost due to color ambiguities. Kuhn et al. [KO*08a] focus on trying to preserve the original image colors. Rasche et al. [RGW05a], instead, try to maintain color distance ratios. Lau et al. [LHM11] preserve both the contrast between color clusters and the transformed image colors.

**Multispectral image fusion** aims to combine a collection of images captured at different wavelengths into a single one, containing details from several spectra. Zhang et al. [ZSM08] and Lau et al. [LHM11] present a method that adaptively adjusts the contrast of photographs by using the contrast and texture information from near-infrared (NIR) image. Kim et al. [JKDB11] show how to use different bands of the invisible spectrum to improve the visual quality of old documents. Süsstrunk and Fredembach [SF10] provide a good introduction to the topic and present, as examples of image enhancements, haze removal and realistic skin smoothing.

**General approaches.** We should stress that most of the methods in the field are targeted to specific applications and lack the generality of a framework that could be applied to different classes of problems. At the same time, there is an obvious common denominator between the aforementioned problems: for example, both color-blind transformations [RGW05b] and color-to-gray conversions [C*10, ZT10, ZF12] can be regarded as mappings to a gamut of lower dimension [G*05]. To the best of our knowledge, only the recent work of [LHM11] introduces a comprehensive approach that works with generic color transformations and easily adapts to different applications.

**Main contribution.** In this paper, we present Laplacian colormaps, a new generic framework for computing structure-preserving color transformations that can be applied to different problems. Our main motivation comes from recent works on Laplacians as structure descriptors [Bd13] and joint diagonalization of Laplacians [E*12, K*13, GB13, BGL13] Using Laplacians as image structure descriptors, we observe that good color transformations should preserve the Laplacian eigenstructure, implying that the Laplacians of the original and color-converted image should be jointly diagonalizable. Employing the relation between joint diagonalizability and commutativity of matrices [GB13, BGL13], we use Laplacians commutativity as a criterion of image structure preservation. We try to find such a colormap that would produce a converted image whose Laplacian commutes as much as possible with the Laplacian of the original image. Since Laplacians can be defined in any colorspace, our approach is generic and applicable to any kind of color conversions (in particular, color-to-gray, gamut mapping, color-blind optimization, etc.). Furthermore, we can work with both global and local colormaps, and use only a tiny portion of the image pixels, leading to an efficient and flexible framework.

We stress that though Laplacian operators have been used in image processing [SKM98, SM00], so far each image has been considered individually. Here, we are interested in common eigenstructures of Laplacians of multiple images, being, to the best of our knowledge, the first application of the results on joint diagonalizability and commutativity of Laplacians in the domain of image analysis.

The rest of the paper is organized as follows: in Section 2, we review the main results related to joint diagonalization and commutativity of matrices. In Section 3 we formulate our optimization problem and discuss its numerical solution. Section 4 shows examples of applications of our framework.
to different problems involving color transformations. Finally, Section 5 concludes the paper. Technical derivations are given in the Appendix.

2. Background

Notation and definitions. We denote by \( \mathbf{A} \) a matrix, by \( \mathbf{a} \) a (column) vector, and by \( a \) a scalar. We denote by

\[
\| \mathbf{A} \|_F = \left( \sum_{ij} a_{ij}^2 \right)^{1/2}; \quad \| \mathbf{a} \|_2 = \left( \sum_i a_i^2 \right)^{1/2}
\]

the Frobenius norm of the matrix \( \mathbf{A} \) and the Euclidean norm of a vector \( \mathbf{a} \), respectively. \( \text{diag}(a_1, \ldots, a_n) \) is a diagonal matrix with diagonal elements \( a_1, \ldots, a_n \). \( \text{diag}(\mathbf{A}) \) is the diagonal matrix obtained by setting to zero the off-diagonal elements of \( \mathbf{A} \), and \( \text{vec}(\mathbf{A}) \) is a column vector obtained by column-stacking of \( \mathbf{A} \).

Let us be given an \( N \times M \) image with \( d \) color channels, column-stacked into an \( NM \times d \) matrix \( \mathbf{X} = [x_1, \ldots, x_{NM}]^T \). The problem of color conversion is creating a new image \( \mathbf{Y} = \Phi(\mathbf{X}) \) with \( d' \) color channels, by means of a colormap \( \Phi : \mathbb{R}^{NM \times d} \rightarrow \mathbb{R}^{NM \times d'} \). In particular, we are interested in parametric colormaps \( \Phi_\theta \), parametrized by an \( n \)-dimensional vector of parameters \( \theta \). In the simplest case, \( \Phi_\theta \) is a global color transformation applied pixel-wise, i.e., each pixel \( x_i \in \mathbb{R}^d \) of the original image is mapped by means of the same \( \Phi_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} \) such that \( \Phi_\theta(\mathbf{x}) = (\Phi_\theta(x_1), \ldots, \Phi_\theta(x_{NM}))^T \) (a simple example is linear RGB to gray mapping, where \( d = 3, d' = 1, n = 3 \) and \( \Phi_\theta(x_i) = \sum_{j=1}^{d} \theta_j x_{ij} \), where in addition we require \( \theta \geq 0 \) and \( \sum_{j=1}^{d} \theta_j = 1 \).

Let \( \{k_1, \ldots, k_s\} \subseteq \{1, \ldots, NM\} \) denote a subset of the image pixel indices, and \( x'_k \) the corresponding colors. When using full size image, \( L = MN \) and \( x'_k = x_k \). If the image is resized by factor of \( s \), \( L = MN/s^2 \) and \( x'_k \) is obtained by averaging the neighbor pixels. Finally, we can also use superpixels [RM03] (with \( x' \) being the average colors therein) to sub-sample the image. Considering these \( L \) pixels as vertices of a graph, we define edge weights (adjacencies) as a combination of spatial and ‘radiometric’ distances,

\[
w_{ij} = -\frac{s^2}{\sigma_x} e^{-\frac{|x'_i - x'_j|^2}{\sigma_x^2}}, \tag{2}
\]

where \( \delta_{ij} \) is the spatial distance between pixels \( k_i \) and \( k_j \), and \( \sigma_x, \sigma_r \geq 0 \) are parameters.\(^\dagger\) Parameter \( \sigma_x \) controls the ‘spatial resolution’; for practical computations, it is usually assumed that \( w_{ij} = 0 \) between spatially-distant (\( \delta_{ij} \gg \sigma_x \)) pixels, which guarantees a sparse structure of the adjacency matrix. We denote by \( K \) the number of non-zero elements in each row of the adjacency matrix. We define the (unnormalized) Laplacian of this graph as a symmetric positive semi-definite \( L \times L \) matrix \( \mathbf{L}_X = \mathbf{D}_X - \mathbf{W}_X \), where \( \mathbf{W}_X \) is the adjacency matrix with elements as in (2), and \( \mathbf{D}_X = \text{diag}(\sum_{j \neq i} w_{ij}) \). In the following, we refer to \( \mathbf{L}_X \) as the Laplacian of image \( \mathbf{X} \).

Since \( \mathbf{L}_X \) is symmetric, it admits an orthonormal eigendecomposition by means of a matrix \( \mathbf{U} \), such that \( \mathbf{U}^T \mathbf{L}_X \mathbf{U} = \mathbf{A}_X \), where the columns of \( \mathbf{U} \) are orthonormal eigenvectors, and \( \mathbf{A}_X = \text{diag}(\lambda_1, \ldots, \lambda_L) \) are the corresponding eigenvalues, sorted in ascending order \( 0 = \lambda_1^X \leq \lambda_2^X \leq \ldots \leq \lambda_L^X \). For simplicity, we assume that there are no repeating eigenvalues, and thus the eigenvectors are defined up to sign. We say that two Laplacians \( \mathbf{L}_X \) and \( \mathbf{L}_Y \) are jointly diagonalizable if they have the same eigenvectors \( \mathbf{U} \), i.e., \( \mathbf{U}^T \mathbf{L}_X \mathbf{U} = \mathbf{A}_X \) and \( \mathbf{U}^T \mathbf{L}_Y \mathbf{U} = \mathbf{A}_Y \). \( \mathbf{L}_X \) and \( \mathbf{L}_Y \) are said to commute if their commutator is \( [\mathbf{L}_X, \mathbf{L}_Y] = \mathbf{L}_Y \mathbf{L}_X - \mathbf{L}_X \mathbf{L}_Y = 0 \).

Image Laplacians as structure descriptors. Laplacians have been successfully used in image processing to guide anisotropic diffusion [SKM98]. Shi and Malik [SM00] showed that a spectral relaxation of the normalized cut criterion for image segmentation boils down to finding the first eigenvectors of an image Laplacian and performing segmentation in the low-dimensional eigensubspace. This approach inspired the popular spectral clustering algorithm [NJW02]. More recently, Bansal and Danilidis [BD13] used the eigenvectors of image Laplacians to perform matching of images taken in different illumination conditions, arguing that Laplacians act as image self-similarity descriptors [SI07].

Applying this idea to color transformations, we can use the similarity of Laplacian eigenspaces as a criterion of image structural similarity. Figure 2 shows three images (original RGB and two decolorized versions thereof, a ‘bad’ and a ‘good’ one) and the first eigenvectors of the corresponding Laplacians. One can see that a good colormap preserves the image structure, which is manifested in the two Laplacians having similar eigenvectors (first and second rows). In particular, if one applies spectral clustering to such images, the resulting segmentation will be similar. Thus, a good (structure-preserving) color transformation \( \Phi(\mathbf{X}) \) would make the corresponding Laplacians \( \mathbf{L}_X \) and \( \mathbf{L}_X(\Phi(\mathbf{X})) \) jointly diagonalizable. In the next paragraphs we show that, in the real case, even approximately jointly diag-

\(^\dagger\) More generally, the ‘radiometric’ part of the adjacency \( w_{ij} \) does not have to work on pixel-wise colors, and one can consider some local features, the simplest of which are patches [WK12].
nalizable Laplacians serve the purpose, giving us the freedom to apply different transformations with different constraints on their parameters.

**Joint approximate diagonalization (JAD)** is a way to find a common eigenstructure of two matrices. Given two matrices $A$ and $B$, one seeks a joint approximate eigenbasis $\tilde{U}$ such that $\tilde{U}^T A \tilde{U}$ and $\tilde{U}^T B \tilde{U}$ are approximately diagonal,

$$J(A, B) = \min \text{off}(\tilde{U}^T A \tilde{U}) + \text{off}(\tilde{U}^T B \tilde{U})$$

where $\text{off}(A) = \sum_{i \neq j} a_{ij}^2$. JAD has been recently applied to jointly diagonalize Laplacian matrices in order to find compatible Fourier bases on graphs [E’12] and surfaces [K’13]. The drawback of this formulation is that both matrices are assumed to be given, while in our problem only one matrix (the original image Laplacian, $L_X$) is given, while the other (the transformed image Laplacian, $L_Y$) has to be found.

**Closest commuting operators (CCO).** Joint diagonalizability is intimately related to matrix commutativity. It is well-known that $A$ and $B$ are jointly diagonalizable if they commute, i.e., $[A, B] = 0$ [HJ90]. It appears that this relation also holds for almost-commuting matrices, in the following sense:

**Theorem 2.1 (Glashoff-Bronstein [GB13])** Let $A, B$ be two $N \times N$ symmetric matrices normalized such that $\|A\|_F = \|B\|_F = 1$. Then,

$$\delta_1(\|A, B\|_F) \leq J(A, B) \leq \delta_2(\|A, B\|_F)$$

where $\delta_1(x), \delta_2(x)$ are functions satisfying $\lim_{x \to 0} \delta_i(x) = 0$; or in other words, almost commuting matrices are almost jointly diagonalizable. Bronstein et al. [BGL13] studied an alternative problem of finding the closest commuting matrices $\tilde{A}, \tilde{B}$ to the given $A$ and $B$,

$$C(A, B) = \min_{\tilde{A}, \tilde{B}} \|\tilde{A} - A\|_F + \|\tilde{B} - B\|_F$$

s.t. $\tilde{A} \tilde{B} = \tilde{B} \tilde{A}$  

Let us summarize the main results of this section, which will motivate our approach described in the following. First, Laplacians can be used as structural descriptors of images. Second, two images having similar structures translates into having the corresponding Laplacians jointly diagonalizable. Third, joint diagonalizability is equivalent to commutativity.

The key idea of this paper is to find such a colormap $\Phi(X)$ that the Laplacian $L_X$ of the input image and the Laplacian $L_{\Phi(X)}$ of the output image commute as much as possible. Due to the relation between approximate commutativity and joint diagonalizability, it will imply that $L_X$ and $L_{\Phi(X)}$ have similar eigenspaces, and thus the underlying images are structurally similar.

### 3. Laplacian colormaps

**Problem formulation.** Let $X$ be a given $N \times d$ original image and $\Phi(X)$ be the desired color-converted $N \times d'$ image. Our goal is to find a set of parameters $\theta$ such that the structures of the images $X$ and $\Phi(X)$ are as similar as possible, where the similarity is judged by the commutativity of the corresponding Laplacians. This leads us to a class of optimization problems of the form

$$\min_{\theta} \mu_0 \|L_X - L_{\Phi(X)}\|_F^2 + \mu_1 \|L_X - L_{\Phi(X)}\|_F^2$$

s.t. constraints on $\theta$.

One can easily recognize in problem (5) a parametric version of the CCO problem (4) with one of the Laplacians fixed. Note that the Laplacian $L_{\Phi(X)}$ is parametrized by a small number of degrees of freedom $n \ll L$, and thus it would be usually impossible to make it exactly commute with the given $L_X$ - hence, unlike the CCO problem, the commutator norm appears as a penalty rather than a constraint.

Additional regularization ($\mu_2$ and $\mu_3$-terms in (5)) is used if we have some ‘nominal’ parameters $\hat{\theta}$ representing a standard color transformation, or if some colors $X_c = (x_1, \ldots, x_p)^T$ should be mapped into some $Y_c = (y_1, \ldots, y_p)^T$ known in advance (for example, in some cases it is important to preserve black and white colors). Finally, depending on the type of the colormap $\Phi$, one may impose some constraints on the parameters $\theta$ (e.g., in linear RGB-to-gray conversion, $\theta \geq 0$ and $\theta^T 1 = 1$).

**Local maps.** Our approach imposes no limitations on the complexity of the colormap $\Phi$; in particular, this map does not have to be global. Let us assume that the source image is partitioned into $q$ (soft) regions, represented by weight vectors $w_1, \ldots, w_q$ of size $N \times 1$, such that $\sum_{i=1}^q w_i = 1$ and $w_i \geq 0$. In each region $i$, we allow for a different colormap $\Phi_i$. Then, the overall colormap is given as $\Phi(X) = \sum_{i=1}^q w_i \Phi_i(X)$, parametrized by $\theta = (\theta_1, \ldots, \theta_q)$. Optimization w.r.t. to the parameters of the local colormap is performed in exactly the same manner as described above.

**Multiple Laplacians.** In some applications like multispectral image fusion, one may wish to impose structural
The Laplacian size \( L \) is a reduction factor. Said differently, the complexity is linear in multiple similarity between the output image and multiple images, \( X_1, \ldots, X_K \) with colorspaces of dimensionality \( d_1, \ldots, d_K \). The input image \( X \) may be one of the \( K \) images or a merged image with \( \sum_{k=1}^{K} d_k \)-dimensional colorspace. In this case, our optimization problem (5) takes the form

\[
\min_{\theta \in \mathbb{R}^d} \sum_{k=1}^{K} \mu_k \| L_{X_k} - L_{\Phi(X)} \|^2_F + \mu_1 \| L_{X_k} - L_{\Phi(X)} \|^2_F \\
+ \mu_2 \| \theta - \theta_0 \|^2_F + \mu_3 \| \Phi(x_c) - Y_c \|^2_F
\]

s.t. constraints on \( \theta \),

where \( \mu_0, \ldots, \mu_K, \mu_{11}, \ldots, \mu_{1K}, \mu_2, \mu_3 \geq 0 \) are constants determining the tradeoff between different penalties.

**Complexity.** The main complexity in the computation of the cost function (5) and its gradient comes from the product of two Laplacian matrices \( L_X L_{\Phi(X)} \), which have the same sparse structure with at most \( K \) non-zero elements in each row. For an \( N \times M \) image, this complexity is \( O((MN)^2 K) \). For an image of size \( N/s \times M/s \), the complexity is thus expected to drop by a factor of \( s^2 \), which encourages us to use resized images for the colormap parameters optimization. However, in practice this would typically be less than \( s^2 \), unlike the above analysis suggests: since resizing also involves local averaging of pixels (e.g. in the simplest 50% subsampling, four pixels become one), the resulting Laplacian is \( s^2 \) times smaller but at the same time \( s^2 \) times less sparse. Overall, the complexity will be \( O \left( \frac{(MN)^2}{s^2} K s^2 \right) = O \left( \frac{(MN)^2}{s^2} K \right), \) i.e., an \( s^2 \) complexity reduction factor. Said differently, the complexity is linear in the Laplacian size \( L \).

**Implementation.** A flow chart of our algorithm is shown in Figure 3, exemplifying the problem of RGB to gray conversion. Given an \( N \times M \) RGB image \( X \), we compute its Laplacian \( L_X \) (this can be optionally preceded by a subsampling procedure, whereby original \( MN \) pixels are reduced to \( L \) using e.g. uniform resizing or superpixels). A parametric transformation \( \Phi_\theta \) is applied to the original image \( X \), resulting in a grayscale image \( \Phi_\theta(X) \), for which the corresponding Laplacian \( L_{\Phi_\theta(X)} \) is computed (undergoing, if necessary, the same subsampling procedure). Optimization of (5) w.r.t. the colormap parameters \( \theta \) is performed; at each step, a new image \( \Phi_\theta(X) \) is computed from the original one.

4. Results and Applications

In this section, we show several applications of our approach for decolorization, image optimization for color-blind people, gamut mapping, and multispectral image fusion, providing extensive comparison to previous works. Due to space limitations, additional results and comparisons appear in the supplementary materials and in [EKB13]. The optimization was implemented in MATLAB, using interior-point method from the Optimization Toolbox. The code used to produce the experiments is available at [http://www.inf.usi.ch/bronstein/EG2014](http://www.inf.usi.ch/bronstein/EG2014).

**Evaluation.** As a quantitative criterion of the colormap quality, we use the root weighted mean square (RWMS) error proposed by [KOF08b], measuring the distortion of relative color distances in two images,

\[
e_i = \left( \frac{1}{NM} \sum_{j=1}^{NM} \frac{(R_y \| x_i - x_j \| - R_X \| y_i - y_j \|)^2}{R_X \| x_i - x_j \|^2} \right)^{1/2},
\]

where \( N \times M \) is the image size, \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R}^d \) denote the \( i \)th pixel of the input and output images, respectively, and \( R_X = \max_{ij} \| x_i - x_j \| \) is the color range of image \( X \). Plotting the pixel-wise RWMS error \( e_i \) as an image allows to see which pixels are most affected by the color transformation. The average \( \frac{1}{NM} \sum_{i=1}^{NM} e_i \) is used as a single number representing the quality of the colormap.

**Settings.** All experiments share a common setup: first of all, RGB values are scaled by 255. Then, unless specified otherwise, we calculate a weighted adjacency matrix according to (2) using all pixels \( (L = MN) \) with fixed 4-neighbors connectivity and parameters \( \alpha = 1, \sigma = 0 \). Default weights for the cost function are \( \mu_0 = \mu_1 = \mu_2 = 1, \mu_3 = 0 \), and \( \theta_0 = 0 \). Parameters \( \theta \) are initialized randomly and normalized such that \( \theta^T 1 = 1 \).

We used the following non-linear parametric colormap:

\[
y_{ik} = \alpha_k + \sum_{j=1}^{d'} \beta_{ikj} x_{ij},
\]

where \( d \) and \( d' \) are, respectively, the number of input and output channels, \( x_{ij} \) is the \( j \)th channel of the \( i \)th input pixel, \( y_{ik} \) is the \( k \)th channel of the \( i \)th output pixel, and \( \theta = (\alpha, \beta_{11}, \ldots, \beta_{d'd'}, \gamma_{11}, \ldots, \gamma_{d'd'}) \) is the \( 2dd' + d' \)-dimensional vector of colormap parameters (note that a simpler linear map with \( dd' \) parameters is obtained by setting \( \alpha = 0, \gamma_{ij} = 1 \); we use the constraint \( \theta \geq 0 \). Furthermore, we distinguish between a *global* map, where the same colormap is applied to each pixel in the image, and a *local* map, where the parametric form of the colormap is the same, but different pixels in the image might have their own vectors of parameters.
Decolorization. In this experiment, we reproduced the benchmark of [Cad08], comparing to previous works [G*05, RGW05b, GD07, NvN07, S*08, LX12] (see Figure 10). In our method, we used default settings and non-linear colormap with $d=3$ and $d'=1$ (7 parameters). Results were evaluated using two different metrics: quantitative (RWMS) and qualitative perceptual evaluation following [Cad08]. In the perceptual evaluation conducted through a Web survey, 124 volunteers were shown the original RGB image together with a pair of its gray conversions, and were asked which of the two results better preserved the original image. Then, we used Thurstone’s law of comparative judgments to convert the 2884 pairwise evaluations into interval z-score scales [Thu27, TG11]. Table 4 provides average RWMS values and z-scores calculated on an 8-images subset of Cadik’s. On average, our approach performs the best w.r.t. both criteria.

Figure 4 shows an example of local colormap using the same non-linear map, but with different θ assigned to groups of pixels obtained by clustering. Parameters for this experiment are $\sigma = 5, \mu_1 = 10, \mu_2 = 0.01$, and $\mu_3 = 10^2$ with constraints on the main color of each cluster. Pixels on the cluster boundaries are finally blended with a gaussian low-pass filter to ensure a smooth transition across clusters.

Color-blind viewers. We model the color distortion of an RGB image $X$ as perceived by a color-blind person by means of a map $\Psi : R^{NM \times d} \rightarrow R^{NM \times d}$. Since $\Psi$ is given and beyond our control, we try to ‘pre-transform’ the original image by means of $\Phi_0 : R^{NM \times d} \rightarrow R^{NM \times d}$ in such a way that the image $(\Phi_0 \circ \Psi)(X)$ appears to the color-blind person as the structure of the original image $X$. In our experiment $\Phi_0$ is the non-linear global map with $d=3, d'=3$ (21 parameters). $\Psi(X) = XA_\Psi$ is a linear transformation, where $A_\Psi$ is a $3 \times 3$ matrix akin to the ones in [K*12], used to replicate the results of [LHM11]:

$$A_\Psi = \begin{pmatrix} 0.8313 & 0.0487 & -0.2286 \\ 0.1991 & 0.8527 & 1.1179 \\ -0.0717 & 0.0715 & 0.1203 \end{pmatrix}$$

We extend our problem formulation so that the transformed image maintains its structure both when seen by a color-blind observer and when seen by a regular observer. In our optimization problem, this translates into requiring the two pairs of Laplacians $L_X, L_{(\Phi_0 \circ \Psi)(X)}$ and $L_X, L_{(\Phi_0)(X)}$ to commute. The cost function is similar to the multiple Laplacians setting (6):

$$\min_{\theta \in R^3} \mu_0 \left( \| L_X - L_{(\Phi_0 \circ \Psi)(X)} \|_F^2 + \mu_1 \| L_X - L_{\Phi_0(X)} \|_F^2 \right) + \mu_1 \left( \| L_X - L_{(\Phi_0 \circ \Psi)(X)} \|_F^2 + \mu_2 \| L_X - L_{\Phi_0(X)} \|_F^2 \right) + \mu_2 \| \theta - \theta_0 \|_2^2 + \mu_3 \left( \| (\Phi_0 \circ \Psi)(X) - Y_c \|_F^2 \right)$$

Figure 5 shows Laplacian colormaps for two different types of color blindness (protanopia and tritanopia). Qualitatively, our result appears to be much closer to the original image compared to [LHM11] (this is especially apparent in the tritanopia case) such that a ‘normal’ viewer sees less distorted colors, while a color-deficient viewer can clearly see the structure structured in the image (digit 6 and different candies) which otherwise would disappear. Quantitatively, we obtain smaller RWMS error, suggesting that our mapping better preserves the original structure of the image, even in those areas that are critical for other approaches.

Gamut mapping is a problem similar to the previous one and applies $\Phi_0$ with the same parameters. Additionally, a transformation $\Psi$ which maps colors from RGB to the XY chromaticity space and a color gamut $G$ (a convex polytope, and in this particular experiment the triangle used in [LHM11] with $R=0.5, 0.32$, $G=0.47$, $B=0.23, 0.18$)) are given. Our goal is to find $\theta$ minimizing the cost (8) subject to $(\Phi_0 \circ \Psi)(X) \subseteq G$, which is imposed as a set of linear constraints. We used the parameters $\mu_0 = 10, \mu_1 = 0.5, \mu_2 = 0.2, \mu_0 = 10, \mu_1 = 0.5$ with colors taken from the original RGB image. Figure 6 compares our results with the outputs of HPMINDE [CIE04] and of the method of Lau et al. [LHM11]. Qualitatively, the output of Laplacian colormaps preserves more details of the original picture (see e.g. the plumage on the red parrot’s head). Quantitatively, our algorithm outperforms the other methods in terms of percentage of out-of-gamut pixels.

Multispectral image fusion uses our non-linear global map with $d > 3$ input channels and $d' = 3$ output channels. We use the cost function (6), with $\mu_0 = \mu_2 = \mu_1 = \mu_2 = 1$ and $\mu_1 = 1.7 \times 10^4$. Constraints are provided to preserve colors for five given image features (dark and light trees, water, mountains, sky). This does not only act as regularization, but also provides us a way to automatically order the three output channels. Our method enhances
the original RGB image while preserving correct colors (e.g. trees on mountains do not present the blue-ish halo that appears in [LHM11]). Figure 9 shows a fusion of four photos of a city with different lighting conditions into a single image, which looks visually plausible. Here, we used $\mu_5 = 0.25$ with reference colors taken from the “Evening” image.

**Performance.** Figure 7 shows the results of a $616 \times 596$ image decolorization, using linear (3 parameters, solid line) and non-linear (9 parameters, dashed line) mapping functions both using Laplacians constructed on the full image, its resized version (to 50% and 25% in each dimensions), and using from 1000 to 250 superpixels. We can see that the computational complexity depends approximately linearly on the number of vertices $L$ used, confirming our theoretical analysis. With the smallest Laplacians computed using 253 superpixels, the optimization of the linear colormap parameters took 0.132 sec, which can be considered real-time performance. At the same time, the quality of the map (RWMS error) is approximately constant, showing that such subsampling strategy does not compromise quality.

5. Conclusions

Laplacian colormaps address the problem of structure-preserving color transformations by relying on Laplacians as image structure descriptors and using Laplacian commutativity as a criterion for structure preservation. Given a parametric colormap, we optimize for the parameters that produce an image whose Laplacian commutes as much as possible with the one of the original image, thus preserving its original structure. Since Laplacians can be defined in any colorspace, our approach can be applied to different kinds of colormaps (global or local, with any number of input and output channels, and where part of the mapping is provided a priori). Moreover, Laplacians can be computed using similarity of local feature descriptors rather than individual pixels colors. Overall, we believe that our results show the promise in the use of Laplacians commutators to measure structure similarity, and seem to be the first application of rather theoretical results on joint diagonalization of matrices to very practical problems in image processing.

**References**


[BD13] BANSAL M., DANIILIDIS K.: Joint spectral correspon-
Figure 6: Gamut mapping results. Odd rows, left-to-right: original image, gamut mapping with method of Lau et al. [LHM11], our approach and HPMINDE [CIE04]. Even rows: gamut alerts for the images above (green shows the out-of-gamut pixels).

Figure 8: Multispectral (RGB+NIR) fusion results.
Figure 9: Fusion of images of four different illuminations into a single RGB image (rightmost).

<table>
<thead>
<tr>
<th>RGB</th>
<th>CIE Y</th>
<th>[G*05]</th>
<th>[RGW05b]</th>
<th>[GD07]</th>
<th>[NvN07]</th>
<th>[S*08]</th>
<th>[LXJ12]</th>
<th>Laplacian</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.14 / -1.59</td>
<td>1.42 / 0.84</td>
<td>2.43 / -0.63</td>
<td>1.50 / 0.18</td>
<td>3.72 / -0.06</td>
<td>2.04 / -0.06</td>
<td>0.92 / 0.38</td>
<td>0.92 / 0.92</td>
</tr>
<tr>
<td></td>
<td>2.18 / -1.05</td>
<td>1.96 / -0.10</td>
<td>1.43 / -1.38</td>
<td>1.35 / 0.86</td>
<td>2.22 / -0.29</td>
<td>1.23 / -0.29</td>
<td>1.47 / 0.52</td>
<td>1.19 / 1.15</td>
</tr>
<tr>
<td></td>
<td>5.01 / -0.55</td>
<td>3.42 / -0.89</td>
<td>3.59 / -0.48</td>
<td>3.44 / 1.41</td>
<td>5.44 / -0.66</td>
<td>5.04 / -0.19</td>
<td>2.90 / 0.50</td>
<td>1.28 / 0.86</td>
</tr>
<tr>
<td></td>
<td>9.27 / -0.57</td>
<td>7.05 / -0.53</td>
<td>7.20 / -0.04</td>
<td>7.28 / 1.45</td>
<td>10.17 / -1.05</td>
<td>9.13 / -1.02</td>
<td>6.30 / 1.01</td>
<td>3.78 / 0.76</td>
</tr>
<tr>
<td></td>
<td>1.25 / 0.29</td>
<td>1.36 / -0.84</td>
<td>1.10 / -0.19</td>
<td>1.23 / 0.10</td>
<td>2.64 / -1.07</td>
<td>1.11 / 0.55</td>
<td>0.96 / 0.28</td>
<td>0.96 / 0.89</td>
</tr>
<tr>
<td></td>
<td>0.97 / 0.27</td>
<td>1.24 / -1.30</td>
<td>0.97 / -0.08</td>
<td>1.02 / 0.61</td>
<td>1.66 / -0.86</td>
<td>1.05 / 0.32</td>
<td>0.80 / 0.22</td>
<td>0.85 / 0.82</td>
</tr>
</tbody>
</table>

Figure 10: Decolorization experiment results. Left: original RGB image, right: grayscale conversion results. Rows 2, 5: RWMS error images and mean RWMS (the smaller the better) / z-score (the larger the better) values. Our Laplacian colormap method performs the best in most cases. Additional results are shown in supplementary materials.
Appendix A: Gradients of the cost function

Let \( L_{\mathbf{q}_b}(\mathbf{x}) = \mathbf{D}_{\mathbf{q}_b}(\mathbf{x}) - \mathbf{W}_{\mathbf{q}_b}(\mathbf{x}) \) be the image Laplacian as defined in (2). We denote by \( |W| \) the number of non-zero elements in the adjacency matrix \( W_{\mathbf{q}_b}(\mathbf{x}) \), and by \( n \) the number of parameters \( \mathbf{q} \) of the colormap, respectively. The non-zero elements \( w_{ij} > 0 \) are indexed as \( \mathbf{w}_0 = \{w_1, \ldots, w_{|W|}\} = \{w_{ij1}, \ldots, w_{ij|W|}\} \). \( \mathbf{q}_b^c : \mathbb{R}^d \to \mathbb{R} \) denotes the \( i \)th channel of the colormap, such that \( \mathbf{q}_b^c(x) = (\mathbf{q}_b^c(x_1), \ldots, \mathbf{q}_b^c(x_n))^T \), and \( \nabla \mathbf{q}_b^c \) is its gradient w.r.t. \( \mathbf{q} \).

We now derive the gradients of the cost function (5). The gradient of the \( \mu_2 \)-term is trivial,

\[
\nabla_{\mathbf{q}}[|\mathbf{q} - \mathbf{q}_b|_2^2] = 2(\mathbf{q} - \mathbf{q}_b).
\]

Denote by \( \mathbf{G}_{\mathbf{q}_b}(\mathbf{x}) \) the matrix of size \( NM \times n \), whose \( j \)th row is the gradient of the \( j \)th channel at the \( i \)th pixel, \( \nabla \mathbf{q}_b^c(\mathbf{x}_j) \), and define \( NMD \times n \) matrix \( \mathbf{G}_{\mathbf{q}_b}(\mathbf{x}) = (\mathbf{G}_{\mathbf{q}_b}^c(\mathbf{x}_1)^T, \ldots, \mathbf{G}_{\mathbf{q}_b}^c(\mathbf{x}_n)^T)^T \). Differentiating the \( \mu_1 \)-term w.r.t \( \mathbf{q} \) gives

\[
\nabla_{\mathbf{q}}[\|\mathbf{q}_b^c(\mathbf{x}_i) - \mathbf{y}_i\|_2^2] = 2 \mathbf{G}_{\mathbf{q}_b}^c(\mathbf{x}_i) (\text{vec}(\mathbf{q}_b^c(\mathbf{x}_i)) - \text{vec}(\mathbf{y}_i)).
\]

The gradients of the first two terms of (5) are obtained by applying the chain rule. First, we differentiate the terms w.r.t \( \mathbf{w}_0 \), obtaining a gradient of size \(|W| \times 1\). Next, we differentiate w.r.t \( \mathbf{q} \). The gradient of the adjacency matrix elements \( w_{ij} \) w.r.t. \( \mathbf{q} \) is

\[
\nabla_{\mathbf{q}}[w_{ij}] = \frac{\partial}{\partial w_{ij}} \left[ \sum_{k=1}^{n} (\mathbf{q}_b^c(x_k) - \mathbf{q}_b^c(x_j)) (\nabla \mathbf{q}_b^c(x_k) - \nabla \mathbf{q}_b^c(x_j)) \right].
\]

The gradient of the commutator (\( \mu_3 \)-term) is:

\[
\frac{\partial}{\partial w_{ij}} \|L_{\mathbf{q}_b}(\mathbf{x}_i) - L_{\mathbf{q}_b}(\mathbf{x}_j)\|_2^2 = -2 \left( \mathbf{O}_1 - L_{\mathbf{q}_b}(\mathbf{x}_i) L_{\mathbf{q}_b}(\mathbf{x}_j)^T \right) - \mathbf{O}_2 - [L_{\mathbf{q}_b}(\mathbf{x}_i), L_{\mathbf{q}_b}(\mathbf{x}_j)] L_{\mathbf{q}_b}(\mathbf{x}_j)^T \right]_{ij}.
\]

The gradient of the \( \mu_1 \)-term is:

\[
\frac{\partial}{\partial w_{ij}} \|L_{\mathbf{q}_b}(\mathbf{x}_i) - L_{\mathbf{q}_b}(\mathbf{x}_j)\|_2^2 = 2 \left( \mathbf{O} + L_{\mathbf{q}_b}(\mathbf{x}_i) L_{\mathbf{q}_b}(\mathbf{x}_j)^T \right)_{ij}.
\]

Here, \( \mathbf{O}, \mathbf{O}_1, \mathbf{O}_2 \) are matrices with equal columns given by

\[
\mathbf{O} = (\text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_1)), \ldots, \text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_n))),
\]

\[
\mathbf{O}_1 = (\text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_1) L_{\mathbf{q}_b}(\mathbf{x}_2)^T), \ldots, \text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_1) L_{\mathbf{q}_b}(\mathbf{x}_n) L_{\mathbf{q}_b}(\mathbf{x}_n)^T)),
\]

\[
\mathbf{O}_2 = (\text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_1) L_{\mathbf{q}_b}(\mathbf{x}_2)^T L_{\mathbf{q}_b}(\mathbf{x}_3)^T), \ldots, \text{diag}(L_{\mathbf{q}_b}(\mathbf{x}_1) L_{\mathbf{q}_b}(\mathbf{x}_n) L_{\mathbf{q}_b}(\mathbf{x}_n)^T L_{\mathbf{q}_b}(\mathbf{x}_n)^T)).
\]

Finally, the gradient of the colormap appearing in the expressions above depends on the choice of the colormap. For all the experiments using the colormap \( \mathbf{q}_b^c(x) \) defined in Section 4, the derivation of the gradient is straightforward. In the experiments simulating color blindness, the colormap is \( \mathbf{q}_b^c \circ \Psi(x) \), whose gradient is given as \( \nabla_{\mathbf{q}_b^c} \circ \Psi(x) = J_{\Psi} \nabla \mathbf{q}_b^c(x) \), where \( J_{\Psi} \) is the Jacobian of \( \Psi \). In our experiments, transformation \( \Psi \) simulating the deficient observer is linear \( \Psi(x) = \mathbf{Ax} \), and thus \( J_{\Psi} = \mathbf{A} \).