

Color to Gray Conversion Using ISOMAP

Ming Cui · Jiuxiang Hu · Anshuman Razdan · Peter Wonka

the date of receipt and acceptance should be inserted later

Received: date / Accepted: date

Abstract In this paper we present a new algorithm to transform an RGB color image to a grayscale image. We propose using non-linear dimension reduction techniques to map higher dimensional color vectors to lower dimensional ones. This approach generalizes the gradient domain manipulation for high dimensional images. Our experiments show that the proposed algorithm generates competitive results and reaches a good compromise between quality and speed.

Keywords ISOMAP, color to gray, color space

1 Introduction

In this paper we investigate how ISOMAP [35], a non-manifold learning technique, can be used for color-image processing. We also present an ISOMAP based framework to map a higher dimensional image to a lower dimensional image, e.g. map a color RGB image to a grayscale image. The problem can be formulated as follows. An $m \times n$ multi-channel image can be seen as a higher dimensional tensor $I_D \in R^{m \times n \times D}$ where each of the mn pixels corresponds to a color vector C_i with D spectral samples. As output of this algorithm we want to map this image to a lower dimensional display range, i.e. a tensor $I_d \in R^{m \times n \times d}$ with all entries constrained to lie between 0 and 1 and $d < D$. In this paper we will consider $D = 3$ and $d = 1$ and map the input image to gray-scale.

We set two goals: 1) The color distances from the input color space can be controlled by the user in the output color space. 2) The algorithm should make use of the dynamic range of the display device to show details

in the image. We will propose an elegant solution that combines both goals in a unified framework.

We are interested in applying manifold learning techniques to the problem at hand. This gives two interesting results: 1) a new operator for image processing; 2) the mapping quality and speed.

One popular technique in the image processing community is to extract the gradient field and then manipulate it to a desirable target. In the end, the target images are reconstructed from the target gradient field [16]. When dealing with multidimensional input, the calculation of the gradient becomes controversial. Our manifold learning approach generalizes the idea to multidimensional data: we first get the matrix of the pairwise distances for the input pixels and then manipulate the distances in the matrix. In the end, we reconstruct an output that preserves the manipulated distances.

The quality and speed performances of our algorithm are compared to several recent approaches, published by Gooch et al. [17], Rasche et al. [27], Grundland et al. [18] and Smith et al [32]. Gooch et al. [17], Rasche et al and [27] are computationally slow. The algorithm by Gooch et al. [17] does not allow higher dimensions because it is intrinsically linked to the $L^*a^*b^*$ color space and the algorithm by Rasche et al. [27] does not scale well to a higher number of spectral samples in an image. By contrast, our solution computes a non-linear mapping by using a linear operator in a sub-manifold of the higher dimensional color space. This approach gives similar visual quality as well as improves computation times and can extend to higher dimensions. Our algorithm is slower than a fixed global mapping, e.g. Smith et al [32]. While such a simple operator can get great results on a large number of images, it is easy to show

that a fixed global mapping can eliminate arbitrarily large features.

Our major contributions are as follows:

- We are the first to apply non-linear manifold learning to the color to gray conversion problem. Our algorithm gives competitive results compared to state-of-the-art algorithms.
- In our RGB to gray mapping algorithm we propose a new way of nonlinearly adjusting the contrast by a single parameter.

One major design decision is if the mapping should be global or local. While most recent tone mapping algorithms favor a local mapping, Rasche et al. [27] argue that a global mapping is important to avoid artifacts when it comes to mapping higher dimensional color vectors to lower dimensional ones. It is worth mentioning that the default implementation of Rasche et al [27] compares every pixel to every other pixel when minimizing the objective function. The authors also suggested an alternative implementation by limiting the comparison to only a small spatial neighborhood for each pixel. This will accelerate their algorithm. However, this also turns the algorithm into a local contrast enhancement operator since widely separated points in original space may be assigned to the same output intensity if surrounded by sufficiently different other color values. Smith et al. [32] also has a local edge sharpening step. In this paper we will present a global mapping algorithm. However, our algorithm can use a local mapping to increase contrast as a post process. As other existing algorithms have the same option, we will not make a potential post process a focal point of this paper.

2 Related Work

There is a large number of techniques to convert a high dynamic range luminance image to a low dynamic range luminance image. These techniques are broken down into local and global methods. Global mappings ensure that identical color values are mapped to identical color values, so that each pixel in an image can be mapped separately [1,37]. Local mappings are typically more complex and slower, however they can adapt the mapping function locally to produce better results [13, 16, 22, 29]. As these methods have several advantages and disadvantages, recent work also focused on combining tone mapping operators [23] and evaluation of tone mapping [21, 26].

In recent years, transforming a color image into a gray scale image attracted the interest of several researchers [4,14,17,18,24,27,28,34,36]. The problem is to

find a lower dimension embedding of the original data that can best preserve the contrast between the data points in original data. These papers are very related to our work and we compare our results against two of them in this paper. The main difficulty of previous work is that they use complex and slow non-linear optimization algorithms. We believe that this is too complex for the problem at hand. In contrast, we want to follow the strategy of manifold learning and first detect a sub-manifold in higher dimensional data before computing a mapping [5, 11, 25, 30, 31, 35]. It worth mentioning that recently two other accelerated methods are proposed and report very good quality. Grundland et al [18] make use of predominant component analysis and accelerate with gaussian pair sampling. Smith et al [32] first use a fast global mapping and then use a local edge sharpening technique based on the laplacian pyramid. We also compare our results to theirs in this paper.

The second similar problem is multispectral and hyperspectral image visualization. Traditionally, these images have been visualized as a cube with a suite of interactive tools [33]. One set of tools allows to extract one spectral band at the time or cycle through spectral bands as an animation. To create RGB images, interactive tools can be used to specify red, green, and blue values as linear combinations of spectral bands. That means an rgb value is computed by a matrix vector multiplication. Along these lines several authors suggest methods how to automatically create linear combinations of spectral bands to define the green, red, and blue color channels of a visualization [12, 20, 38, 39]. In this paper we compare our results to two such methods, Jacobson et al. [20] and visualization based on PCA [38]. Recent investigation suggest that nonlinearity exists in hyperspectral data [19]. Actually ISOMAP has been adopted for hyperspectral image visualization in [2, 3]. We believe it is interesting to extend their work to color to gray image conversion. A faster visualization strategy for hyperspectral visualization was proposed by Cui et al [8]., but their method cannot be directly applied to the color to grayscale problem.

There is a larger number of general dimensionality reduction algorithms in the literature. Prominent examples are ISOMAP [35], Local Linear Embedding(LLE) [30], Laplacian Eigenmap Embedding [5]. ISOMAP is a special version of multidimensional scaling, which uses geodesic distance instead of Euclidean distance between the points. LLE tries to preserve the local linear structure of the original point set and casts it as an eigenvalue problem. Laplacian Eigenmap Embedding formulates the problem as a spectral graph cut and also solves it as an eigenvalue problem. In recent years, more advanced versions of manifold learning algorithms are proposed.

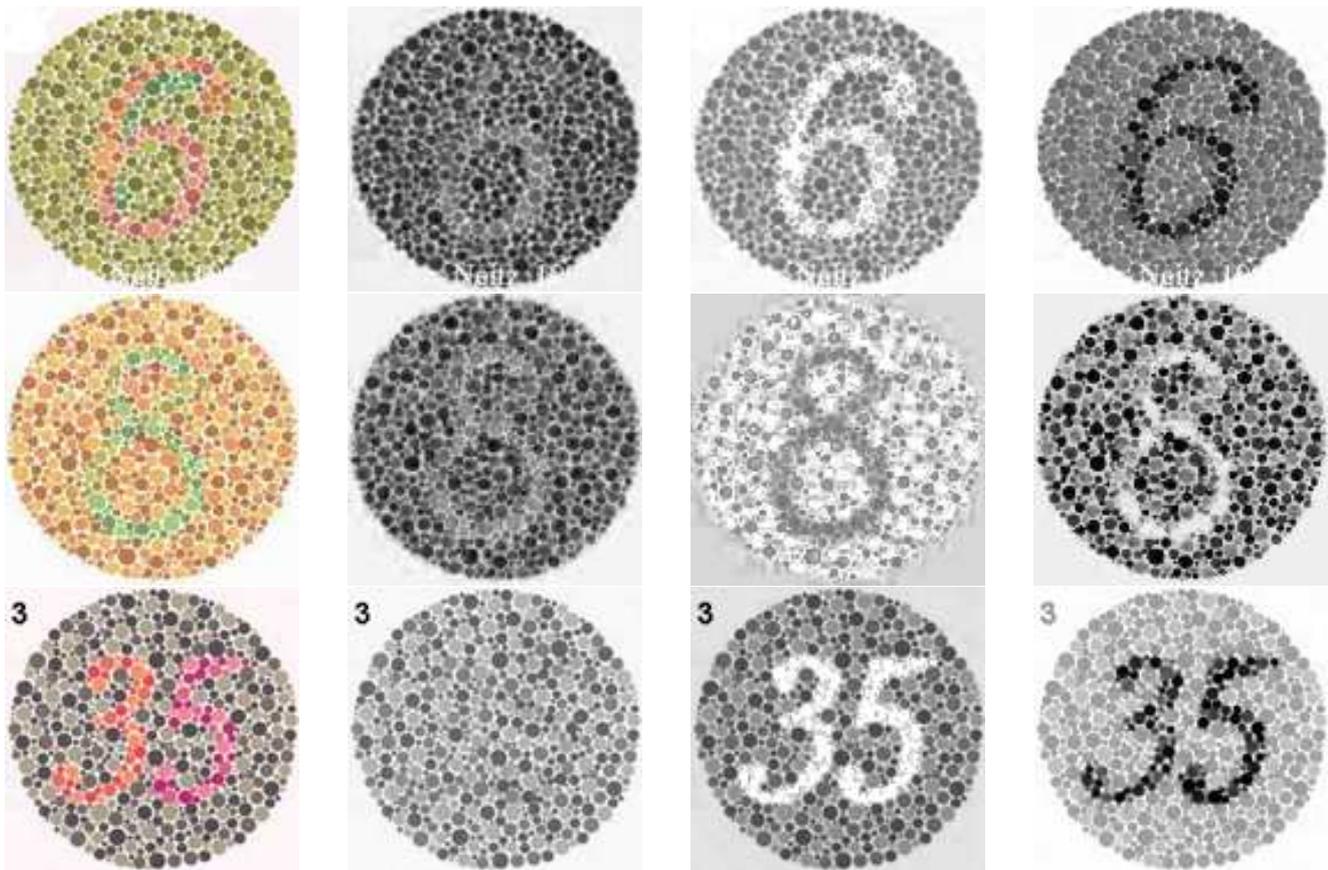


Fig. 1 From left to right: original image, pca mapping, color2gray mapping, and our results.

These include Hessian Eigenmap Embedding [11], Conformal Maps [31], and Diffusion Maps [25]. These methods are usually computationally more expensive.

3 Overview

3.1 Algorithm Goals

We formally state the problem as follows. The input to the algorithm is an $m \times n$ image as tensor $I_D \in R^{m \times n \times D}$ where each of the $N = mn$ pixels correspond to a color vector C_i with D spectral samples. The output of this algorithm is a tensor $I_d \in R^{m \times n \times d}$ where each of the N pixels corresponds to a color vector c_i with d spectral samples and all entries are constrained to lie between 0 and 1. For color to gray conversion $D = 3$ and $d = 1$. There is a one to one correspondence between a color vector (pixel) C_i and c_i .

Our first goal is to find a global mapping that preserves the pairwise distances between all input pixels. This goal can be formalized as finding a mapping that minimizes E :

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (||c_i - c_j|| - dist(C_i, C_j))^2 \quad (1)$$

E can also be presented in a matrix form:

$$E = \frac{1}{2} ||M_c - M_C||_F \quad (2)$$

where F denotes the Frobenius norm, this equation is in matrix form. M_c and M_C are both matrices. $M_c(i, j) = ||c_i - c_j||$ and $M_C(i, j) = dist(C_i, C_j)$.

The second goal is making use of the dynamic range of the display to show image details. This goal is partially in conflict with the first goal and difficult to qualify in a formula, but we found a consistent way to integrate the second goal with the first goal by modeling a distance function $dist(C_i, C_j)$ that provides some user control of the output. It is very important that we only make consistent modifications. For example, a local tone mapping operator can produce colorful images, but the original meaning of the input is not preserved. This can be very counterproductive for visualization, because pixels are no longer comparable. Similarly, a

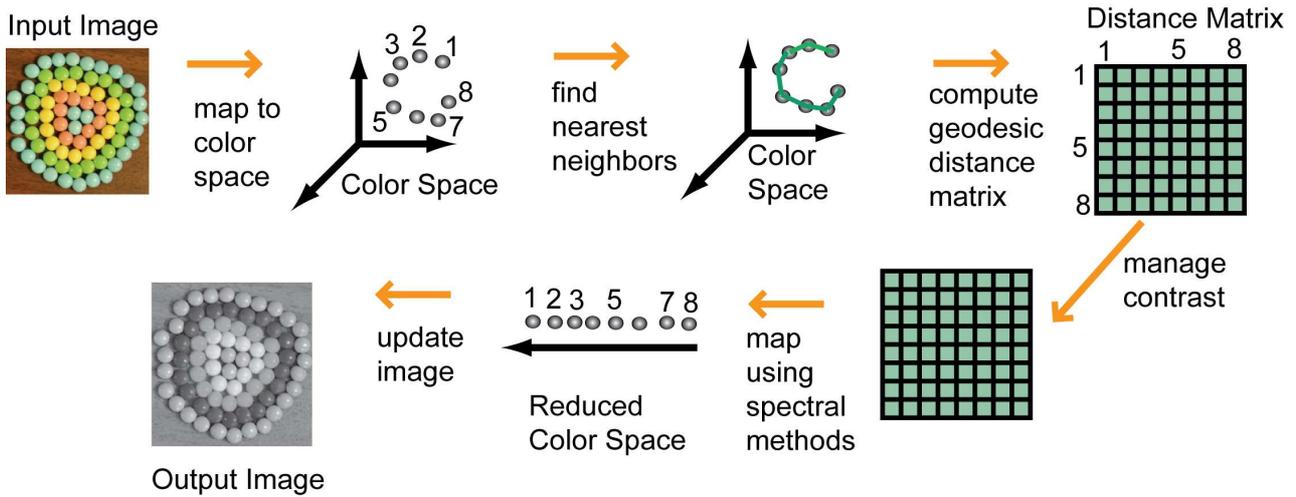


Fig. 2 Overview of our algorithm:

global operator such as histogram equalization in all color bands sometimes introduce artificial features that are not present in the data set .

3.2 Algorithm Overview

An overview of our algorithm is shown in figure 2. It computes a nonlinear mapping. In general, a nonlinear mapping is much better at adapting to the structure of the data and it was therefore also used in previous approaches. The algorithm includes the following stages:

Color Space Preprocessing: We take an input image and consider each pixel as a higher dimensional color vector. This gives us a set of vectors in a higher dimensional color space. If the input image has RGB color vectors we additionally map all pixels from RGB to $L^*a^*b^*$ color space.

Sub-manifold Detection: Find a sub-manifold in higher dimensional space, by computing geodesic rather than Euclidian distances. This stage includes finding nearest neighbors, computing a geodesic distance matrix, and managing contrast by transforming the matrix. The output of this stage is a distance matrix defining pairwise distances between all pairs C_i and C_j .

Optimized Mapping: Find an optimized mapping from higher to lower dimensional color vectors. At this stage each color vector C_i is mapped to a lower dimensional color vector c_i based on a matrix decomposition. This operation is very fast and finds a global optimum.

Color Space Postprocessing: The color mapping can be used to construct a lower dimensional image I_d . Postprocessing can include local (a gradient domain poisson solver [16]) or global (histogram equalization) tone mapping operators.

Acceleration Strategy: While the above algorithm steps define a working algorithm, we need to accelerate the algorithm and reduce memory consumption by using a subsampling strategy. The main idea is to subsample the rows of the matrix D_C .

4 An Introduction to ISOMAP

In this section we give a brief introduction to ISOMAP, a very successful strategy for manifold learning that was proposed by Tenenbaum et al. [35]. ISOMAP in essence is a special version of the classical multidimensional scaling (Classical MDS) algorithm [6].

4.1 Classical MDS Algorithm

Classical MDS [6] provides a solution for equation 2. Since a global optimum cannot be found for 2, classical MDS does not minimize the F-norm of the difference matrix $M_c - M_C$ in equation 2 directly. Instead, it minimizes the difference of two transformed matrices. The transform first computes an element-wise square of a matrix and then centers it. The centering operator τ for a matrix M can be computed by $\tau(M) = -HMH/2$, and $H = I - 1/N * O$ with O being a matrix of all ones. If we denote the element-wise square of M_c and M_C as M_c^2 and M_C^2 respectively, we can express the objective of the transformed minimization problem as:

$$E = \|\tau(M_c^2) - \tau(M_C^2)\|_F \quad (3)$$

Geometrically, we are now minimizing the pairwise angular distances instead of the pairwise Euclidean distances. The benefit we gain from this transform is that

the global optimum of equation 3 can be computed in close form. Let us denote $\lambda_1, \lambda_2, \dots, \lambda_d$ as the largest d eigenvalues of matrix $\tau(M_C^2)$ and v_1, v_2, \dots, v_d as their corresponding eigenvectors. Then the d -dimensional output c_i is computed as [9].

$$c_i = \begin{bmatrix} \sqrt{\lambda_1} \cdot v_{1i} \\ \sqrt{\lambda_2} \cdot v_{2i} \\ \dots \\ \sqrt{\lambda_d} \cdot v_{di} \end{bmatrix} \quad (4)$$

4.2 The ISOMAP Algorithm

In classical MDS, how to calculate $dist(C_i, C_j)$ is left for the user to decide. In the ISOMAP algorithm, Tenenbaum et al. proposed using the geodistance between the input data points for $dist(C_i, C_j)$: the input dataset is treated as a graph. Each input point in the original D -dimensional space is a node in the graph and connected to its k nearest neighbors (k is a parameter provided by the user). The distance between two points C_i and C_j is calculated as the shortest path between the two corresponding nodes in the graph.

The ISOMAP algorithm has two computational bottlenecks [10]: First, for a graph that has N nodes, it takes $(O(N^2 \log N))$ to find all pairwise shortest path in the matrix D_C using Dijkstra's algorithm. Second, we need to solve the eigenvalue problem for the N by N matrix $\tau(M_C)$, which takes $O(N^3)$. The overall complexity is $O(N^3)$. For a medium size image with size 300 by 300, the total number of pixels, which is N , equals 90000 and the algorithm is very slow.

4.3 The Landmark ISOMAP Algorithm

In [9], an accelerated version of ISOMAP is proposed called landmark ISOMAP. The new algorithm starts by selecting only a small fraction of the whole input point set called landmark points L_1, L_2, \dots, L_n . If we denote n as the number of landmark points then usually we pick $n = \lceil N * 0.02 \rceil$. The original ISOMAP algorithm is run on the n landmark points to get a skeleton for the output in the d -dimensional output space. The rest of the output is embedded into the skeleton by projecting to the first d principle axes of the landmark points. We denote the squared pairwise distance matrix for the landmark points as L_C^2 and λ_i and v_i as the eigenvalues and eigenvectors of $\tau(L_C^2)$ as before. We further define δ_i as a column vector of the squared distances from C_i to all the landmark points and δ_L to be the mean of all

the column vectors of L_C^2 . Now we can express the j th component of c_i as

$$c_{ij} = -\frac{1}{2} \frac{v_j}{\sqrt{\lambda_j}} (\delta_i - \delta_L) \quad (5)$$

Since only the pairwise distances between the landmarks and the remaining points are needed for the interpolation, the cost for Dijkstra's algorithm is reduced to $O(nN \log N)$. The ISOMAP algorithm on landmark points requires $O(n^3)$. Since $n \ll N$, the overall complexity is reduced to $O(N \log N)$.

5 Algorithm Details

In the following we explain in detail how the landmark ISOMAP algorithm is used for our color to gray mapping problem.

5.1 Color Space Preprocessing

Given an RGB image with m rows and n columns, we can interpret each pixel as a color vector C_i in a D -dimensional color space yielding $N = mn$ color vectors. If we are working with RGB images as input we transform each of the vectors C_i into $CIE L^* a^* b^*$ color space. This transformation is useful, because the Euclidean distance between the pixels in $L^* a^* b^*$ is similar to the visual difference perceived by human eyes. The L^* coordinate refers to the luminance and the a^* and b^* coordinates describe the position of the pixel in a two-dimensional chromatic space. The positive a^* -axis points towards red and the negative a^* -axis points towards green. Similarly, the positive b^* -axis points towards yellow and negative b^* -axis points towards blue blue. As the color space preprocessing is optional, we use the same variables C_i to denote the input and the output of this stage. $L^* a^* b^*$ Color space preprocessing only works when the input is an RGB image. Figure 3 shows an example color space. For multispectral images it is important to drop the spectral bands that are destroyed by atmospheric water vapor. Additionally, we provide the option to scale individual spectral bands (we did not use this feature in our results).

5.2 Submanifold Detection

This part of the algorithm computes pairwise distances between two color vectors C_i and C_j . There are several choices for a distance function $dist(C_i, C_j)$ from equation 1 and we want to propose a different one from previous work. For example, Rasche et al. [27], use the

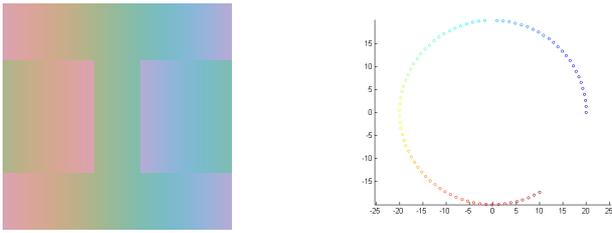


Fig. 3 Left: input image. Right: color distribution in $L^*a^*b^*$ color space. This figure shows how an image is mapped to $CIEL^*a^*b^*$ color space. The shown color distribution will require a non-linear mapping to ensure a meaningful conversion to gray scale. Please note, that no line can be found in color space so that a color projection onto the line results in a useful gray scale conversion.

simple Euclidian distance function (in $CIEL^*a^*b^*$ color space) and reduce the problem to a multi-dimensional scaling problem that requires non-linear majorization techniques to find a local optimal solution. Color2Gray [17] define a more complex distance function as a combination of the luminance difference and the difference of projection on a certain direction in the chromatic space. This distance function leads to a non-linear optimization problem that the authors solve with a conjugate-gradient method.

Following the ISOMAP framework described in section 4, we propose to use a transformed geodesic distance $dist(C_i, C_j) = f(geodesic(C_i, C_j))$ between two color vectors. The geodesic distance is computed in a manifold spanned by the color vectors and brings a significant advantage. The strength of this algorithm is that we can compute a *global* optimum directly with multidimensional scaling. However, please note that the geodesic distance computation provides a mapping of one color space to another that is still non-linear and is able to adapt to the structure of the data.

The distance computation has three steps: First, we need to construct a graph by computing the k -nearest-neighbors(KNN) of each input point C_i and connect each point with its k nearest neighbors with an edge weighted by their mutual distances. In our experiments, k is usually set to 10 50. Second, we need to compute a geodesic distance matrix by computing shortest distances from all the landmark points L_i to the rest of all points C_i in the graph. Third, we can manipulate the geodesic distances to provide some user control to improve the contrast in the final image.

An implementation detail worth mentioning is that the graph constructed by the KNN algorithm might not be connected. For example, k input points can be tightly clustered and form a clique of the graph, so that they will be isolated from the rest of the point set. Since we need to know the distance matrix from the landmark points to all the points and the entries of the

matrix cannot be infinity, we use a simple and fast solution to alleviate this problem. We find the dominant component in the graph that has the largest number of points. For any other component, we add an edge connecting the two closest points between this component and the dominant component.

5.3 Contrast Management

Our contrast management is motivated by a strategy common in high dynamic range image tone mapping. The input image is usually first transformed to a new domain in which the manipulation becomes easier. Then the output image is recovered from the manipulated image in the new domain. In [16] the input is transformed into the gradient domain, then the gradient is manipulated and the output is recovered by solving a PDE. In [15] the input is transformed into the wavelet domain, then the wavelets coefficients are manipulated and the output is recovered using the inverse wavelet transform. Similarly, we transform the input into the pairwise distances domain, then we manipulate the distances, and the output is recovered by the ISOMAP algorithm.

The manipulation can be done by nonlinearly scaling the $geodesic(C_i, C_j)$ distance function with another function f . This allows to enhance contrast, while still using a distance preserving mapping. We chose a simple function $f(x) = x^\lambda$ that allows the user to control one parameter in the mapping. The output of our contrast management step is a distance matrix D_C with entries $x_{ij} = f(geodesic(C_i, C_j))$. This matrix can be used to directly project color vectors in a lower dimensional color space as described in the next subsection. We illustrate the power of the parameter λ using two examples shown in figure 4 and 5. Intuitively, the parameter can either enhance local contrast (small distances) and reduce global contrast (large distances), or enhance global contrast and reduce local contrast. Please note that the parameter controls geodesic distances.



Fig. 4 Right: Input Image: Middle: a small $\lambda = 1$ reveals a 70 Right: A bigger $\lambda = 4$ reveals a 29.

In figure 6 we demonstrate the usefulness of non-linear mapping and geodesic distances. When we compare the results of PCA and our algorithm, we can see

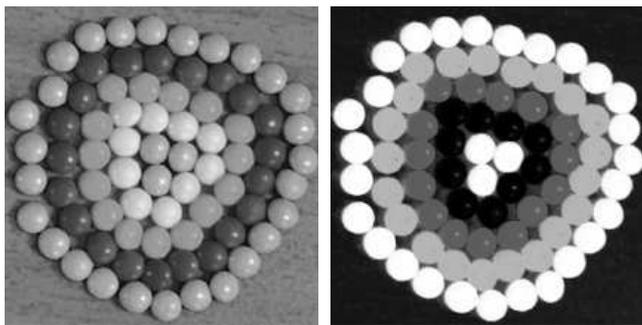


Fig. 5 Possible settings for the candy example from figure 11. Right: a small $\lambda = 0.8$ preserves the details of the image well and the spherical shape and the highlights of the candy are well visible. Right: A bigger $\lambda = 3$ reduces local contrast and the spheres look like flat circles.

that PCA is unable to detect the nonlinear structure of the problem and does not show the two rectangles. While it is not reasonable to assume that the colors of a real world example with thousands of colors are aligned in this way, this nonlinear structure problem would appear many times in subsections of the color space.

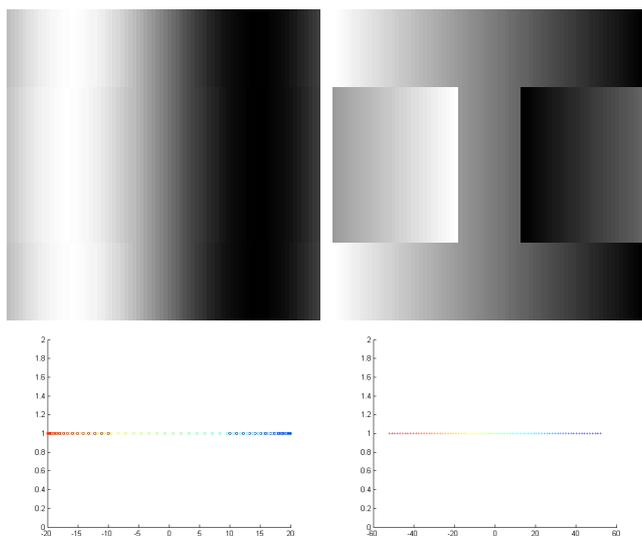


Fig. 6 Top Left: The PCA result on the input image in figure 3. Top Right: our result. Bottom Left: We show how the original colors are projected to 1D for the PCA result. Bottom Right: We show how the original colors are projected to 1D for our result.

5.4 Landmark Selection

In the original paper [9] the author proposes to select the matrix rows randomly. The author suggests that a more sophisticated clustering technique might have disadvantages because the clustering problem is

domain specific and the computational cost of clustering is high. While these arguments are reasonable for a general algorithm, our experiments show that clustering can improve the results of our algorithm. First, we found that this is fairly risky to make a random selection, as parts of the color space with few color vectors can receive no or insufficient samples. In graphics applications it might be interesting to ignore outliers, but for visualization applications outliers often convey important and meaningful information. Second, we need to build the kd-tree to accelerate the KNN algorithm so we can reuse it for the computation of more uniformly distributed landmarks. To select landmarks we can set a parameter s that specifies the subsampling factor. We traverse the kd-tree depth first and stop the traversal if a node contains less than $s \times N$ points. We select the centroid of all the points contained in the kd-tree node as landmark.

5.5 Color Space Postprocessing

After we get a set of color vectors c_i , we can assemble the low dimensional image. If the output is grayscale we linearly map the range of all output intensity values to the range of $[0, 1]$. Additionally, we implemented histogram equalization and a gradient domain tone mapping algorithm [16] as potential post process to enhance contrast. However, while we found that these algorithms can make certain images to look nicer, they make it hard to compare with other algorithms. First, these post-processing operations are in conflict with the goals of the original algorithm and second the comparison with other methods would be influenced. We also omit a comparison to Smith et al [32]. They basically use an edge sharpening operator to achieve their results in difficult cases, but we consider this a potential post process.

6 Results

6.1 Visual Comparison

The authors in [7] gave an exhaustive comparison of the state-of-the-art color to gray algorithms. We run our algorithm through the same set of test images with a fixed parameter $\lambda = 0.4$. The number of nearest neighbors K is fixed to 50. There are more complicated algorithms to dynamically decide K according to the neighborhood of a particular point, but we opted not to use these computationally more expensive approach. In the results, we found that with the help of λ we are able to enhance the details in the original image in a novel way that no

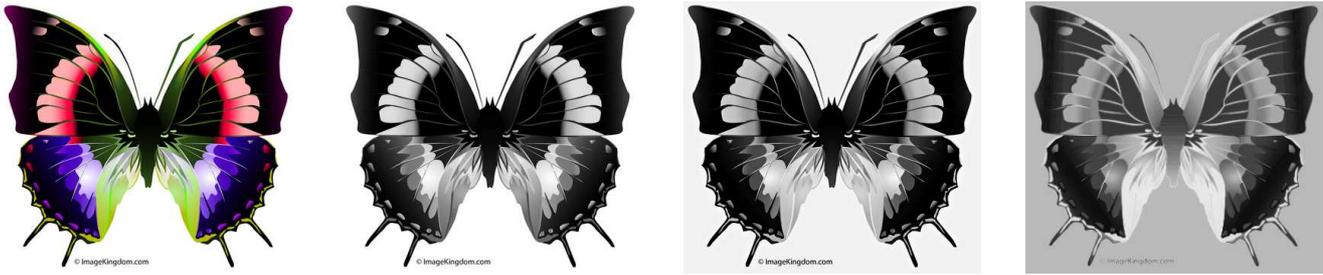


Fig. 7 From left to right: original color image; decolorize [18]; smith08 [32]; ours, $\lambda = 0.4$.



Fig. 8 From left to right: original color image; smith08 [32]; color2gray [17]; ours, $\lambda = 0.4$.



Fig. 9 From left to right: original color image; smith08 [32]; CIEY [14]; ours, $\lambda = 0.4$.

other algorithms can achieve. In figure 7, 8, 9, and 10 we show some examples. In each case we picked the best two algorithms ranked in [7] and compare to our results. In figure 7, it can be observed that our result can distinguish the color difference between the two antennas. Also the body colors are better stratified than the other two. The outlines in the upper parts of the wings are also the clearest among the three. In figure 8, we can see that in the middle right part of the tree, there are three blocks of red leaves. Our method can highlight these three red blocks. In figure 9, both other two methods mapped the red pedals to similar colors as the green leaves. In contrast, our method mapped the red pedals to the brightest color in the whole image, which is satisfactory. In figure 10, at the bottom left of the original images, there are some faint circles that only a careful observe can find. All other methods neglected these circles. In contrast, our algorithm highlights these circle. This might be helpful if these circles are real features. At the same time, we also want to point out the

watch face and the printed page seem to be too bright. We might also enhance noise and artifacts instead of real features that are desirable. Results on other test images can be found in the supplementary material of the paper. It worth noting that, we are not doing an objective user study for the results like [7]. Therefore, we are not claiming that our results are always better than other algorithms. We emphasize that our algorithm can highlights some features of the original images that other algorithms might omit. It is not always the case that our results are the most satisfactory. For example, in figure 14, our method does not distinguish the orange-highlighted lines and the pink-highlighted lines very well. More comparisons are shown in the supplementary material. In some cases, our results show salt-and-pepper noise artifacts. We find that the artifacts depend on the parameter k in the k -nearest-neighbors search. We will further discuss this in section 6.3.



Fig. 10 From left to right: original color image; smith08 [32]; decolorize [18]; ours, $\lambda = 0.4$.

6.2 Timing Comparison

We implemented our algorithm in Matlab on a 3.6Ghz Xeon processor. The algorithm is of average complexity and takes about 500 lines of code including the display, input, and output routines. While Matlab greatly helps the simplicity of implementation, a complete C++ implementation would be much faster. However, since our algorithm speed is already very competitive, we opted against further low level optimizations and porting our code to C++.

We used color images proposed in previous work to compute an RGB color to gray scale mapping. We selected a set of six input images that were picked as examples in previous work, so that we could compare directly to the images produced by other algorithms. In figure 1 we compare the visual results of color blind tests, a linear mapping by using PCA, a mapping using Color2gray [17], a mapping using the algorithm by Rasche et al. [27], and our algorithm. We used the C++ implementation from the original authors for Color2gray algorithm and Rasche et al. algorithm. We did not use any GPU acceleration of all algorithms.

In the table 1 we show timings of the results. Even though we compare our results in a matlab implementation we can outperform the implementation speed of the other algorithms.

	C2G [17]	Rasche [27]	Ours	image size
Number 6	57	n/a	6	124*120
Number 35	52	n/a	6	128*125
Number 8	46	n/a	4	120*120
Candy	494	400	32	212*218
Flower	2100	330	86	311*300
Sunrise	1060	n/a	42	295*212

Table 1 Comparison of computation times in seconds of Color2Gray, Rasche et al., and our algorithm. Each row of the table are results for one dataset whose name is in the leftmost column.

Figure 11, 12, and 13 we show more visual comparisons. We would argue that our visual result for the candy image is stronger than the competing algorithms.

For the flower and sunrise image we can see advantages and disadvantages in all algorithms.

6.3 Parameter Selection

We found that the parameter k in the k-nearest-neighbors search algorithm plays an important role for the final results. The larger the value of k , the higher the quality of the results. An example is shown in figure 14. When k is not big enough, there might be some noticeable salt-and-pepper noise artifacts in the result. However, increasing k will significantly increase the computation time. For example, in figure 15, when $k = 15$, the computation time is 10.37s. When $k = 50$, the computation time is 12.53s. When $k = 200$, the computation time is 17.50s. Larger k will also make the distance matrix less sparse and significantly increase the storage requirement. For our experiments we fix k to be 50 to balance the speed and the quality.

7 Discussion

In the following we discuss the results and the comparison to other algorithms.

Local vs Global operator: Our mapping algorithm is a global operator. It means that for two input points C_i and C_j , if their values are the same, they will always be mapped to the same grayscale value c_i in spite of their spatial locations in the input image. This is in contrast to other local mapping operators, which will distort the grayscale values to enhance local contrast. We did not include a local enhancement step for two reasons: first, the local enhancement cannot be naturally incorporated in the ISOMAP framework; second, we believe in some cases global operators might be desirable. For example, if we want to do tone mapping on an output grayscale image or mutual information based registration on an output image pair, global operators will be preferable.

Control of the output: Similar to Color2Gray [17] and Rasche et al. [27] we allow for some control of the output. Similar to Color2Gray we are able to select a

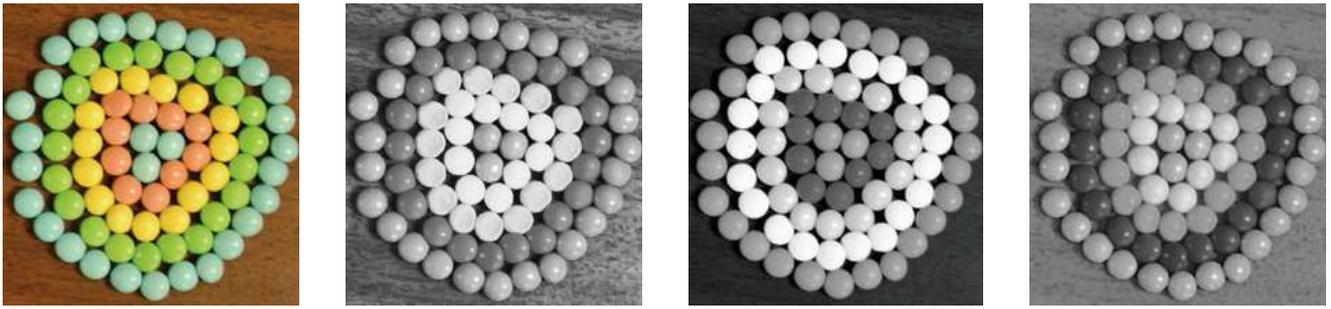


Fig. 11 From left to right: the original image, color2gray mapping [17]; Rasche et al. mapping [27]; ours, $\lambda = 0.8$.



Fig. 12 From left to right: the original image; color2gray mapping [17]; Rasche et al. mapping [27]; ours, $\lambda = 2.0$.

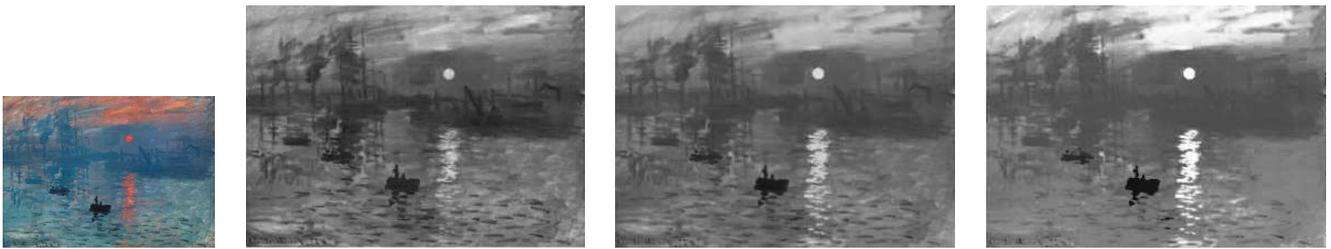


Fig. 13 From left to right: the original image; decolorize [18]; color2gray [17]; ours, $\lambda = 2.0$.

direction in $L^*a^*b^*$ color space and decide whether to map this direction to a lighter or darker color. However, we do not have an equivalent for luminance consistency constraints in Rasche et al.'s algorithm. The idea of luminance consistency is to enforce that the relative order of luminance of colors of similar gamut is enforced. While this control mechanism sounds reasonable in the description, we were not able to verify its importance in the test images and the influence of luminance consistency constraints was also not evaluated in the original paper. In contrast to previous approaches we provide a parameter λ for contrast management.

Linear vs. non-linear optimization: We think the strength of this algorithm is that it uses a linear operation to map colors from a higher dimensional color space to a lower dimensional one. While the overall mapping is still non-linear, the non-linear aspect is due to the sub-manifold detection and geodesic distance computation that precedes the actual mapping step. Our optimization can be computed directly by spectral

matrix decomposition and leads to a global optimum. This part of the algorithm is very stable and does not have the many problems of non-linear optimization. In contrast, previous algorithms [17,27] use non-linear optimization. While non-linear optimization is a powerful tool, it is also very hard to setup and it is not really possible to know if the proposed solution is close to the global optimum.

Submanifolds in color images: Similar to previous work our algorithm is beneficial if the color distribution in the original RGB image contains enough complexity. Some color images are simple enough so that the simple conversion with Photoshop is sufficient. Even though our algorithm is especially suitable for color distributions that exhibit local or global lower dimensional manifolds we found the output to be meaningful for all input images.

Limitations: The algorithm has several limitations and challenges. First, the sub-manifold detection algorithm uses a k -nearest-neighbor algorithm (KNN) to

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

The luminance generated by a physical device is generally **not a linear function of the applied signal**. A conventional CRT has a **power-law response to voltage**; luminance produced at the face of the display is approximately proportional to the applied voltage raised to the 2.5 power. The numerical value of the exponent of this power function is colloquially **known as gamma**. This nonlinearity must be compensated in order to achieve correct reproduction of luminance.

As mentioned above (*What is lightness?*), human vision has a nonuniform perceptual response to luminance. If luminance is to be coded into a small number of steps, say 256, then in order for the most effective perceptual

Fig. 14 First row, from left to right: original image; CIEY [14]; decolorize [18]. Second row, comparison of different settings of parameter k for our method. From left to right: $k = 15$; $k = 30$; $k = 50$.



Fig. 15 From left to right: original color image; Our method with $k = 15$; $k = 50$; $k = 200$; all the other parameters are fixed.

find neighbors for each color vector in color space. KNN has some known disadvantages: a fixed number of K might not work well on all dataset; KNN search becomes slow in high dimensions. It might be possible to improve results using an adaptive KNN sampling algorithm. This typically comes at the cost of implementation speed. It might actually be more useful to take the opposite approach and use a fast approximate k nearest neighbor algorithm instead. The second challenge is the memory consumption of the algorithm. Part of the problem arises from our implementation in matlab. However, it would be worthwhile to explore more aggressive subsampling and clustering strategies for rows and columns.

8 Conclusions

In this paper we explained the ISOMAP algorithm and used it for color image to grayscale conversion. We cast the problem into a dimension reduction problem that has a simple parameter for users to control the level of contrast enhancement naturally. The speed of the algorithm is fast and the quality is competitive with respect to the state-of-the-art global color to gray algorithms. There are several interesting avenues for future work. First, we want to implement an out of core algorithm for the mapping, so that a high quality reference solution can be computed for very large images. Second, based on the current solution, we want to improve the quality of the mapping. We are also interested in algorithms for

pre-processing textures and shaders in computer games to assist color blind users.

References

1. Alessandro Artusi, Jiří Bittner, Michael Wimmer, and Alexander Wilkie. Delivering interactivity to complex tone mapping operators. In Per Christensen and Daniel Cohen-Or, editors, *Rendering Techniques 2003 (Proceedings Eurographics Symposium on Rendering)*, pages 38–44. Eurographics, Eurographics Association, June 2003.
2. Charles M Bachmann., Thomas L. Ainsworth, and Robert A.Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):441–454, March 2005.
3. Charles M Bachmann., Thomas L. Ainsworth, and Robert A.Fusina. Improved manifold coordinate representations of large scale hyperspectral scenes. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2786–2802, October 2006.
4. R. Bala and K. M. Braun. Color-to-grayscale conversion to maintain discriminability. In *SPIE Conference Series*, volume 5293 of *SPIE Conference Series*, pages 196–202, December 2003.
5. Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.
6. I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, Second edition, 2005.
7. Martin Čadík. Perceptual evaluation of color-to-grayscale image conversions. *Comput. Graph. Forum*, 27(7):1745–1754, 2008.
8. Ming Cui, Anshuman Razdan, Jiuxiang Hu, and Peter Wonka. Interactive hyperspectral image visualization using convex optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 47(6):1673–1684, JUN 2009.

9. V. de Silva and B. Tenenbaum. Sparse multidimensional scaling using landmark points. *Technical Report*, 2004.
10. V. de Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction, 2003.
11. D. Donoho and C. Grimes. Hessian eigenmaps: locally linear embedding techniques for high dimensional data. *Proc. of National Academy of Sciences*, 100(10):5591–5596, 2003.
12. N. Cai S. Younan N. Du, Q. Raksuntorn. Color representation and classification for hyperspectral imagery. *IGARSS*, pages 537–540, Aug 2006.
13. Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, 21(3):257–266, July 2002.
14. Mark D. Fairchild, editor. *Color Appearance Models*. Wiley-IST, 2005.
15. Raanan Fattal. Edge-avoiding wavelets and their applications. *ACM Trans. Graph.*, 28(3):1–10, 2009.
16. Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 249–256, New York, NY, USA, 2002. ACM Press.
17. Amy A. Gooch, Sven C. Olsen, Jack Tumblin, and Bruce Gooch. Color2gray: salience-preserving color removal. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 634–639, New York, NY, USA, 2005. ACM Press.
18. Mark Grundland and Neil A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recogn.*, 40(11):2891–2896, 2007.
19. Tian Han. and D.G. Goodenough. Investigation of nonlinearity in hyperspectral remotely sensed imagery : a nonlinear time series analysis approach. *IGARSS*, pages 1556–1560, July 2007.
20. N.P. Jacobson and M.R. Gupta. Design goals and solutions for display of hyperspectral images. 43(11):2684–2692, November 2005.
21. Patrick Ledda, Alan Chalmers, Tom Troscianko, and Helge Seetzen. Evaluation of tone mapping operators using a High Dynamic Range display. *ACM Transactions on Graphics*, 24(3):640–648, July 2005.
22. Yuanzhen Li, Lavanya Sharan, and Edward H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph.*, 24(3):836–844, 2005.
23. Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 646–653, New York, NY, USA, 2006. ACM Press.
24. Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. A perceptual framework for contrast processing of high dynamic range images. *ACM Trans. Appl. Percept.*, 3(3):286–308, 2006.
25. Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, Cambridge, MA, 2006.
26. Sung Ho Park and Ethan D. Montag. Evaluating tone mapping algorithms for rendering non-pictorial (scientific) high-dynamic-range images. *J. Vis. Commun. Image Represent.*, 18(5):415–428, 2007.
27. Geist R. Rasche, K. and J. Westall. Re-coloring images for gamuts of lower dimension. *Computer Graphics Forum*, 24(3):423–432, 2005.
28. Karl Rasche, Robert Geist, and James Westall. Detail preserving reproduction of color images for monochromats and dichromats. *IEEE Comput. Graph. Appl.*, 25(3):22–30, 2005.
29. Erik Reinhard, Mike Stark, Peter Shirley, and Jim Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2002 Annual Conference)*, 21(3):267–276, 2002.
30. Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
31. Fei Sha and Lawrence K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 784–791, New York, NY, USA, 2005. ACM Press.
32. Kaleigh Smith, Pierre-Edouard Landes, Joëlle Thollot, and Karol Myszkowski. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2), apr 2008.
33. Randal Smith, editor. *Analyzing Hyperstectral Images with TNTmips*. Microimages, 2006.
34. D.A. Socolinsky and L.B Wolff. Multispectral image visualization through first-order fusion. *IEEE Transactions on Image Processing*, 11(8):923–931, 2002.
35. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.
36. Bruce A. Thomas, Robin N. Strickland, and Jeffrey J. Rodriguez. Color image enhancement using spatially adaptive saturation feedback. In *ICIP '97 Volume 3*, page 30, Washington, DC, USA, 1997. IEEE Computer Society.
37. Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Comput. Graph. Appl.*, 13(6):42–48, 1993.
38. J.S. Tyo, A. Konsolakis, D.I. Diersen, and R.C. Olsen. Principal-components-based display strategy for spectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 41(3):708–718, March 2003.
39. J. Wang and C.I. Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1586–1600, June 2006.