

A User-Assisted Approach to Visualizing Multidimensional Images

Jason Lawrence, Sean Arietta, Michael Kazhdan, Daniel Lepage, Colleen O'Hagan

Abstract—We present a new technique for fusing together an arbitrary number of aligned images into a single color or intensity image. We approach this fusion problem from the context of Multidimensional Scaling (MDS) and describe an algorithm that preserves the relative distances between pairs of pixel values in the input (vectors of measurements) as perceived differences in a color image. The two main advantages of our approach over existing techniques are that it can incorporate user constraints into the mapping process and it allows adaptively compressing or exaggerating features in the input in order to make better use of the output's limited dynamic range. We demonstrate these benefits by showing applications in various scientific domains and comparing our algorithm to previously proposed techniques.

Index Terms—multidimensional images, visualization techniques, dimensionality reduction, multidimensional scaling, physical sciences and engineering, life and medical sciences

1 INTRODUCTION

Multidimensional images are common in many fields including remote sensing, medicine, biology, astronomy, computer vision, computer graphics, and digital photography. They are the result of recording a sequence of scalar-valued images, each sensitive to the physical world in different ways, in a common raster grid. Commonly, these component images correspond to different regions of the electromagnetic spectrum (Figure 1, left).

General-purpose tools for visualizing multidimensional images are critical for analyzing this important type of dataset and a common approach is to compute a single color or intensity image that attempts to capture their most salient features [2], [3], [4]. However, this process represents an inherent loss of information since the number of displayable bands (e.g., three for a color image) is generally less than the number of channels in the input. Moreover, it is often the case that certain types of data (e.g., tissue density measured by a CT scan) have no natural relationship to the colors and intensities visible to the human eye. As a result, there is no single correct solution to this problem and different algorithms perform this mapping in different ways. Nevertheless, researchers have identified a number of properties that a useful visualization ought to have [5], [6]. Chief among these

is consistency: *pixels in the visualization that have similar values may be interpreted as having similar responses in the underlying data and those with different values may be interpreted as having different responses*. Other desirable properties include using a color palette that takes into account the human visual system, preserving strong edges that occur within any single input band, and being efficient to compute.

Researchers have developed many different image fusion techniques optimized for different objectives [4]. A popular approach is to use some type of dimensionality reduction (DR) algorithm such as Principal Component Analysis (PCA) [7] to compute a consistent visualization that preserves as much of the variance in the input as is possible in a lower-dimensional projection (Figure 1, middle). Other automatic techniques involve merging features at different spatial scales using, for example, a Laplacian pyramid [8] or wavelet decomposition [9], or superimposing the spatial gradients of the individual component images from which a single grayscale image may be extracted by solving a Poisson equation [10].

Of these classes of techniques, dimensionality reduction offers a uniquely elegant way of taking advantage of a multidimensional output colorspace. However, previously proposed algorithms suffer from two key drawbacks that we attempt to address. First, although these algorithms excel at capturing large-scale features in the input they attenuate features that occur at finer scales. For example, although the differences between the terrain and water are clearly captured by PCA in Figure 1, subtler variations *within* these regions such as the turbulent swirls of sediment in the water are far less perceptible. Second, current algorithms do not allow a user to intuitively influence the selection of colors during the mapping process.

- J. Lawrence, S. Arietta, D. Lepage, and C. O'Hagan are with the Department of Computer Science at the University of Virginia, 151 Engineer's Way, Charlottesville, VA, 22901.
E-mail: {jdl, sma2t, dl8ym, cohagan}@cs.virginia.edu
- M. Kazhdan is with the Department of Computer Science at Johns Hopkins University, 224 New Engineering Building, 3400 N. Charles St., Baltimore, MD, 21218.
E-mail: misha@cs.jhu.edu

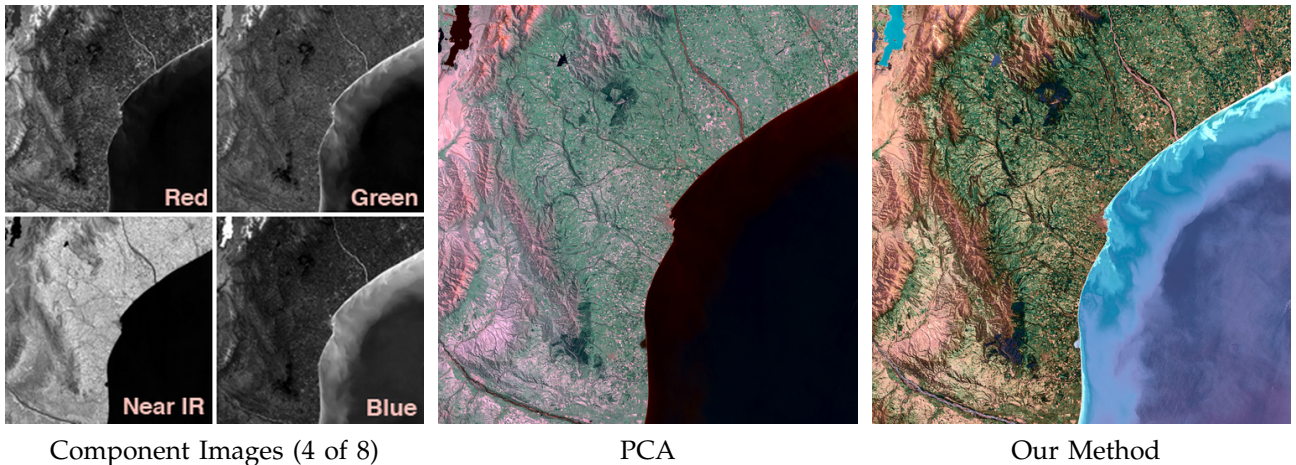


Fig. 1. A multispectral aerial image taken over the coast of New Zealand [1]. **Left:** Four of eight bands in the input. In clockwise order starting from the upper left, the first three images correspond to the standard visible region (R,G,B) and the fourth measures near infrared. **Middle:** Projection of the input onto its first three principal components interpreted as coordinates in the CIELAB color space. **Right:** The result of our scaling algorithm which maintains a more realistic color scheme and exaggerates fine-scale details such as the turbulence along the coastline.

However, domain experts have well-established color schemes and visualization practices and it would be desirable to accommodate user constraints within an otherwise automatic technique. We present an algorithm that retains the benefits of dimensionality reduction while allowing the user to participate in the mapping process and provides a way of preserving features in the input that occur over a wide range of scales.

We approach this image fusion problem from the context of Multidimensional Scaling (MDS) [11], a general method for embedding a high-dimensional point set in a lower-dimensional space so as to preserve interpoint distances. We consider a general non-linear formulation of MDS that can accommodate soft constraints on the solution. Based on a thorough analysis of existing MDS techniques, we propose a practical algorithm inspired by work done in the graph drawing community [12], [13]. Specifically, we conduct the underlying optimization within a low-dimensional subspace of all possible solutions that is constructed to contain a wide range of useful visualizations. We further accelerate this optimization by considering only a sparse set of pairwise distances in a way analogous to existing “landmark” methods [14], [15]. Finally, we present a modification to our basic algorithm that allows exaggerating fine-scale details by modifying the target pairwise distances. Although our technique does not explicitly consider edge information in the component images, our results show that it is effective at preserving these important features as well (Figure 1, right). We evaluate our algorithm using a variety of datasets from remote sensing, medicine, biology and astronomy and compare its performance to state-of-the-art methods.

2 RELATED WORK

The task of computing an informative color or intensity visualization of a multidimensional image arises in many settings ranging from analyzing hyperspectral images in remote sensing to decolorization (or “color to gray”) methods developed in computer graphics. In contrast to the many domain-specific techniques described in the literature, the solution we present is general in that it can be tailored to apply to datasets in different fields. We demonstrate this fact by showing results in medicine, remote sensing, astronomy, and biology. We have structured the following discussion of prior work based on the underlying numerical algorithm. To the best of our knowledge, our approach is the first one that allows incorporating user constraints into the mapping process and gives explicit intuitive control over attenuating or exaggerating features in the input.

Segmentation: The pixel values in a multiparameter image can be quantized into a displayable number of colors using unsupervised segmentation algorithms [16], [17]. The main drawback of these approaches is that they may fail to preserve intra-cluster variation and the process of assigning colors to clusters either is a difficult manual process or relies on heuristics.

Dimensionality Reduction: A common strategy for analyzing high-dimensional datasets is to project them onto a lower-dimensional space where they can be visualized directly. Techniques including PCA and Projection Pursuit [18] compute a linear projection of the data, each optimizing a different criterion, and are commonly used to visualize multiparameter and hyperspectral images [5], [7], [19], [20], [21], [22]. A related set of methods perform a non-linear projection

of the data. This includes Self-Organizing Maps [23] that have been used in this context [24], [25] and the work of Kim et al. [26] that applies Locally Linear Embedding (LLE) [27] to project the input onto a reduced basis. Although non-linear methods are naturally more flexible than linear methods, both classes of techniques struggle to accurately depict inputs that contain a large range of values and they do not provide intuitive control over the choice of colors. A related set of methods are decolorization or “color to gray” techniques developed in the computer graphics community [28], [29], [30]. A method similar to our technique is that of Rasche et al. [29] which applied a similar MDS algorithm, although their formulation did not allow arbitrary color constraints nor did they explore non-linear transformations to the target distances for the purpose of exaggerating details in the input.

Multiscale and Gradient-Domain Methods: An alternative approach is to fuse together features in the component images that occur at different spatial scales [2]. The Laplacian pyramid [31] is well suited for this task and has been used to visualize multiparameter images [8]. However, this process is known to cause blocking artifacts and unwanted halos. Li et al. [9] perform a similar fusion in the wavelet domain, by retaining the maximum wavelet coefficient across the input bands at each pixel. A related approach is due to Scheunders [32] which fuses together *multiscale edges* [33] extracted from each component image.

Although there is no canonical orientation for the direction of maximum contrast at a pixel in a multiparameter image like there is for scalar-valued images [34], a related set of techniques approximate a single gradient direction and magnitude at each pixel based on an analysis of the second fundamental form [10], [35]. This allows computing a grayscale image that has a similar gradient field by solving a Poisson equation.

While these two classes of techniques have the advantage of explicitly preserving spatial gradients in the input in the resulting visualization they have two key drawbacks. First, they produce scalar-valued intensity output. These results are often colorized in practice using an ad-hoc process [10]. Second, they do not explicitly enforce a consistent mapping—two pixels in the output with similar values may have significantly different responses in the underlying input. Dimensionality reduction methods, on the other hand, are better able to utilize a multidimensional output colorspace and produce consistent mappings.

Tone-Mapping and Detail Exaggeration: Our approach is also related to tone-mapping methods [36], [37], [38], [39] designed to compress high-dynamic-range images to a low-dynamic-range display format. Many multidimensional images contain a wide range of values (e.g., astronomical datasets that include bright stars and dim interstellar objects within the

same image). In these cases, we are faced with the problem of effectively mapping these values to a color image with a significantly smaller range of allowable values. Another set of related techniques allow exaggerating fine-scale details in a set of images [40], [41]. We show how our basic algorithm can be modified to allow a similar exaggeration of features in a multidimensional image to more effectively utilize the limited dynamic range of the output.

3 PROBLEM FORMULATION

In this work, we describe a method for mapping a high-dimensional image, represented by a large number of scalar-valued bands, to a low-dimensional image. In the simplest formulation, the goal of such a mapping is to generate a low-dimensional image whose pixel values are *consistent* with the pixel values of the high-dimensional input. That is, the mapping should preserve (to the extent possible) the relative distances between pairs of pixel values in the input.

In this work, we consider a more general problem that supports user-guided mapping in two ways. First, the method should support soft constraints on the color of the output pixels, allowing a user to specify that the non-visual data should be rendered with a domain-specific color palette. Second, the method should support non-linear transformations of distance values, allowing the user to highlight detail by dampening target distances in input ranges that are of low interest and amplifying target distances in important ranges.

An example of such an application is shown in Figure 2. In this example, the input consists of the four bands shown in images (a) and (b) (RGB+UV) and the goal is to generate a three-band (RGB) image which best captures the data in the input. Generating a color image that best preserves the input distances results in an image capturing the variation in the input data but failing to preserve the visual information, shown in (c). Using the input color values as a soft constraint, our system solves for an image whose pixel variations are consistent with the input data and whose color palette is faithful to the original color values, shown in (d). Finally, by adaptively remapping target distances, we obtain an image which enhances detail by compressing target distances for large differences in input data values and inflating target distances for small difference values, shown in (e).

Formally, the goal of our work is to define a mapping of an image with H bands and N pixels to a lower dimensional image with L bands of equal spatial resolution. Denoting by \mathbf{R} the $N \times H$ input matrix of data values and by \mathbf{S} the $N \times L$ output matrix of color values, we aim to solve for the row

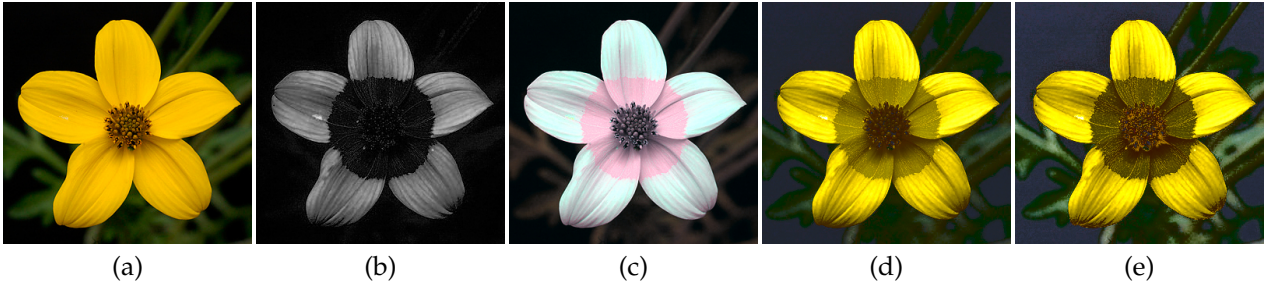


Fig. 2. Overview of our visualization method. This example shows a 4D multispectral image of a *Coreopsis* flower that consists of (a) three visible RGB bands and (b) one ultraviolet (UV) band. Using SVD to map these 4D pixel values to 3D CIELAB coordinates results in (c) an image that conveys the structure in this dataset, but uses an arbitrary color palette. (d) Our method allows for the preservation of the input colors by interpreting the visible bands as soft constraints on the multidimensional mapping process. (e) An important benefit of our method is that it also allows exaggerating details in these datasets by adaptively scaling pairwise distances in the input. Images © Bjørn Rørslett/NN/Nærfoto www.naturfotograf.com.

vectors of \mathbf{S} , $\{\mathbf{s}_1, \dots, \mathbf{s}_N\} \in \mathbb{R}^L$, that minimize:

$$\tau(\mathbf{S}) = \underbrace{\sum_{i < j} w_{ij} (\delta_{ij}^{\mathbf{R}} - \|\mathbf{s}_i - \mathbf{s}_j\|)^2}_{\text{Distance Consistency}} + \underbrace{\sum_i w_i \|\mathbf{s}_i - \bar{\mathbf{s}}_i\|^2}_{\text{Value Constraints}} \quad (1)$$

In this equation, the quantity $\tau(\mathbf{S})$ is referred to as the *stress* of the solution \mathbf{S} [42]. Here, the soft constraints on the pixel values are incorporated on the right, with $\bar{\mathbf{s}}_i$ denoting the prescribed value at pixel i and w_i denoting the weight of the constraint. The distance constraints are incorporated on the left, with $\delta_{ij}^{\mathbf{R}}$ denoting the target distance between the value of \mathbf{s}_i and the value of \mathbf{s}_j and w_{ij} denoting the weight of the associated distance constraint.

3.1 Unconstrained Stress with Euclidean Distances

In the special case where the pixel values are unconstrained ($w_i = 0$) and the target distances are Euclidean and uniformly weighted ($w_{ij} = 1$ and $\delta_{ij}^{\mathbf{R}} = \|\mathbf{r}_i - \mathbf{r}_j\|$), Equation 1 permits a closed-form solution computed from the Singular Value Decomposition (SVD) of the Gram matrix of \mathbf{R} , a technique referred to as classical metric MDS [11]. However, using this technique in practice requires addressing the fact that N is often very large.¹ A common approximation strategy is to consider only a subset of the pairwise distances during the optimization. Two popular algorithms, Landmark MDS (LMDS) [14] and Pivot MDS [15], explicitly embed only a small subset of K points (where $K \ll N$) and then place the remaining $N - K$ points with respect to these fixed locations. The method of Rasche et al. [29] for computing grayscale versions of color images used a technique based on LMDS to handle large images.

1. As a point of comparison, a 512×512 image would require taking the SVD of a matrix with 262K rows and columns.

3.2 Incorporating Constraints and General Target Distances

Our approach, outlined in the following section, performs non-linear optimization to minimize stress [43], [44], [45]. There are two main advantages to this method. First, it allows exploration of a larger portion of the solution space in each iteration, requiring fewer iterations to converge as compared to force-directed approaches [46]. Second, in contrast to classical scaling techniques, it can accommodate the more general formulation of stress in Equation 1.

4 OPTIMIZATION STRATEGY

The challenge in solving for the image \mathbf{S} that minimizes stress is that the expression in Equation 1 results in a non-linear optimization (due to the appearance of $\|\mathbf{s}_i - \mathbf{s}_j\|$ in the distance consistency constraint). To address this, we minimize the stress through an iterative process known as *majorization* [42], obtaining a sequence of images $\{\mathbf{S}^k\}$ with strictly non-increasing stress values, $\tau(\mathbf{S}^{k-1}) \geq \tau(\mathbf{S}^k)$.

Like the Gauss-Newton method, stress majorization proceeds by locally fitting a quadratic (positive semi-definite) energy to the stress at the current estimate, and obtains the next estimate by solving for the minimum of the quadratic energy. The method terminates when the difference in the stress values of two successive estimates falls below a given threshold. The difference between these approaches is that by choosing a majorizing energy (i.e. the value of the quadratic energy is never smaller than the actual stress), the next estimate is guaranteed not to “overshoot”, resulting in estimates $\{\mathbf{S}^k\}$ with monotonically decreasing stress.

While we defer the details of the construction to Appendix A, we remark that since the majorizing energy is quadratic, minimizing the energy is equivalent to solving a symmetric linear system. In practice, the

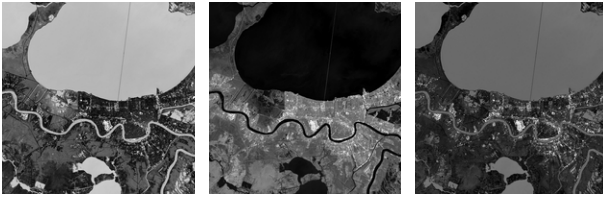


Fig. 3. Three (of fifty) basis vectors computed from an 8-band multispectral aerial image of New Orleans [1]. Euclidean distances are encoded as grayscale intensities so that pixel values similar to the associated cluster center are dark and dissimilar pixel values are white. Projecting the data into this basis preserves pairwise distances.

system that we solve is of the form:

$$(\mathbf{L} + \Lambda)\mathbf{S}^{k+1} = \mathbf{L}^k \mathbf{S}^k + \Lambda \bar{\mathbf{S}}, \quad (2)$$

where \mathbf{L} is a Laplacian matrix whose entries are defined by the distance weights, Λ is the diagonal matrix of constraint weights, and \mathbf{L}^k is a Laplacian matrix whose entries adjust for the relationship between the prescribed target distances and the distances realized by the previous solution \mathbf{S}^k .

4.1 Subspace Restriction

Equation 2 requires solving a dense linear system of equations with NL unknowns, a prohibitively expensive task even for small images. To overcome this limitation, we first adopt an idea proposed within the graph drawing community [12], [13] and restrict the solution to lie within a carefully chosen subspace. This decouples the size of the linear system from the image resolution, making this approach feasible for large datasets.

Formally, we restrict each N -dimensional band (column) in the $N \times L$ solution matrix \mathbf{S}^k to lie within the subspace spanned by M orthonormal column vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_M\}$. Therefore

$$\mathbf{S}^k = (\mathbf{v}_1 \cdots \mathbf{v}_M) \begin{pmatrix} u_{11}^k & \cdots & u_{1L}^k \\ \vdots & \ddots & \vdots \\ u_{M1}^k & \cdots & u_{ML}^k \end{pmatrix} = \mathbf{V}\mathbf{U}^k, \quad (3)$$

where \mathbf{V} is an $N \times M$ matrix constructed from these basis vectors and \mathbf{U}^k is a $M \times L$ matrix that contains the coordinates of the solution \mathbf{S}^k within this basis.

As we explain in the following section, each basis vector \mathbf{v}_i is itself an image derived from the input (Figure 3). This forces each band in the solution to be expressed as a linear combination of M such images and allows rewriting Equation 2 as

$$(\mathbf{L} + \Lambda)\mathbf{V}\mathbf{U}^{k+1} = \mathbf{L}^k \mathbf{S}^k + \Lambda \bar{\mathbf{S}}. \quad (4)$$

We multiply both sides by $\mathbf{V}^T(\mathbf{L} + \Lambda)^T$ to obtain the normal equations

$$\mathbf{V}^T(\mathbf{L} + \Lambda)^T(\mathbf{L} + \Lambda)\mathbf{V}\mathbf{U}^k = \mathbf{V}^T(\mathbf{L} + \Lambda)^T(\mathbf{L}^k \mathbf{S}^k + \Lambda \bar{\mathbf{S}}), \quad (5)$$

a linear system with ML unknowns, in practice $M \sim 50$.

Although restricting the solution to a low dimensional subspace reduces the size of the linear system that must be solved in each iteration, two important issues remain. First, there is the task of defining a suitable basis \mathbf{V} . Second, computing the products in Equation 5 requires an infeasible N^2M operations. We describe our solutions to both of these problems in the following sections.

4.2 Basis Construction

We require a low-dimensional basis that still contains a wide range of accurate and informative visualizations. We follow a procedure similar to that outlined by Harel et al. [12] for constructing \mathbf{V} . We first separate the input pixel values into M clusters using a standard k -means algorithm (clustering the pixels based on their multi-band values and ignoring spatial proximity). For each cluster center $\mathbf{q}_i \in \mathbb{R}^H$, we construct the column vector \mathbf{v}_i , with $v_{ij} = \|\mathbf{q}_i - \mathbf{r}_j\|$. That is, \mathbf{v}_i is the “image” of the distance of pixels in \mathbf{R} from \mathbf{q}_i .² Three basis vectors are shown in Figure 3 that were computed from a multispectral image of New Orleans [1]. The first is obtained by choosing a cluster center representing populated land regions, the second is obtained by choosing a cluster center representing water, and the third is obtained by choosing a cluster center representing less densely populated areas.

Recall that our formulation requires these basis vectors to be orthonormal. We use the Gram-Schmidt process to accomplish this. We also force these basis vectors to be orthogonal to $\mathbf{1} \in \mathbb{R}^N$. This eliminates the translational ambiguity from the solution space and guarantees that the coefficient matrix in Equation 5 is positive definite.³

To understand why a basis constructed in this manner can express a range of useful visualizations, consider the vectors shown in Figure 3. Note that each basis vector encodes a “view” of the data from the perspective of the associated cluster center. It can be shown [12] that nearby pixel values remain nearby and distant pixel values remain distant under projections into this basis. The amount of error due to this projection is proportional to the maximum distance between any one pixel value and its closest cluster center. Because the k -means algorithm is designed to minimize this maximum distance, this error is expected to be small. We present results that confirm the effectiveness of this basis in Section 5.

2. Note that this approach implicitly assumes that the distances, $\delta_{ij}^{\mathbf{R}}$, are representative of the distances between the pixels in the original image \mathbf{R} so that the basis defined using Euclidean distances $\|\mathbf{q}_i - \mathbf{r}_j\|$ should be effective for capturing the variation represented by the non-Euclidean distances described by $\delta_{ij}^{\mathbf{R}}$.

3. Technically, this assumes the graph Laplacian matrix has no disconnected components which is true for our construction described in Section 4.3.

4.3 Sparse Approximation of the Laplacian

Although performing the optimization in a reduced subspace limits the size of the linear system that must be solved, computing the matrix products in Equation 5 remains impractical. To address this, we force the Laplacian matrices \mathbf{L} and \mathbf{L}^k to be sparse by selecting a set of “landmark” pixels P where $|P| = K$ and defining the weights as

$$w_{ij} = \begin{cases} w_{ij} = 1 & \text{if } i \in P \text{ or } j \in P \\ w_{ij} = 0 & \text{otherwise} \end{cases}.$$

These K pixels are also chosen using a k -means clustering algorithm. This sparse assignment of weights reduces the complexity of computing $(\mathbf{L} + \Lambda)\mathbf{V}$ from $O(N^2M)$ to $O(NKM)$ and computing $(\mathbf{L}^k\mathbf{S}^k + \Lambda\tilde{\mathbf{S}})$ to $O(NKL)$. This approach is analogous to existing landmark versions of classical MDS such as Pivot MDS [15] that minimize an approximation to stress by considering a subset of the pairwise distances in Equation 1. The fact that we carry out this optimization in a restricted subspace further improves stability even for very small K [45] as this acts as a regularization operator since many degenerate embeddings are simply not expressible in \mathbf{V} .

4.4 Implementation

Figure 4 gives pseudocode of our complete algorithm along with the asymptotic running time of each major component. Lines 1-3 must be performed only once per dataset and the results are stored on disk. The products in lines 4, 5 and 9 are performed on the GPU using NVIDIA’s CUDA programming framework. In each case, the result is divided into 16×16 submatrices to form a thread block and each thread computes a single entry.

Equation 5 is solved using a standard LU factorization method in LAPACK implemented in Intel’s Math Kernel Library. Only the relatively small $M \times L$ coordinate matrix \mathbf{U} must be transferred between the GPU and CPU in each iteration. The other matrices reside in GPU memory, including the reduced image

Stress Majorization		
1	$\mathbf{V} = \text{ComputeBasis}(\mathbf{R}, M)$	$O(NM)$
2	$\mathbf{V} = \text{GramSchmidt}(\mathbf{V})$	$O(NM^2)$
3	$\mathbf{P} = \text{SelectLandmarks}(\mathbf{R}, K)$	$O(NK)$
4	Precompute $\mathbf{A} = (\mathbf{L} + \Lambda)\mathbf{V}$	$O(NKM)$
5	Precompute $\mathbf{A}^T\mathbf{A}$	$O(NM^2)$
6	Initialize \mathbf{S}	$O(NL)$
7	Repeat until convergence:	
8	$\mathbf{L}^k = \text{LocalLaplacian}(\mathbf{S}, \delta^{\mathbf{R}})$	$O(NK)$
9	Compute $\mathbf{B} = \mathbf{A}^T(\mathbf{L}^k\mathbf{S} + \Lambda\tilde{\mathbf{S}})$	$O(NL(M + K))$
10	Solve $\mathbf{U} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{B}$	$O(M^3)$
11	Compute $\mathbf{S} = \mathbf{V}\mathbf{U}$	$O(NLM)$
12	Compute stress from Eq. 1	$O(NK(L + H))$
13	Return \mathbf{S}	

Fig. 4. Pseudocode for our algorithm.

\mathbf{S} , allowing efficient transfer to the framebuffer for display. We compute the stress in Equation 1 after each iteration and terminate the algorithm when this changes by less than 1%, typically within 20 iterations.

5 EVALUATION

To determine the practical efficacy of our approach, we consider three different evaluations. First, we consider the performance of our method in optimizing raw stress, where the values of the output image are not constrained and the target distances are defined in terms of Euclidean distances between pixel values in the input. In this scenario, traditional approaches can be used to solve for the low-dimensional output image, allowing for a direct comparison to related work. Next, we consider the qualitative contribution of incorporating constraints and non-Euclidean distances into the optimization process. Finally, we provide a quantitative evaluation of our method by presenting a simple metric for evaluating visualization fidelity which we use to compare results obtained with several different methods.

5.1 Raw Stress: No Constraints and Euclidean Distances

We compared our algorithm to Principal Component Analysis (PCA), Pivot MDS [15] and Glimmer [46]. Although PCA is designed to preserve variance instead of pairwise distances, it is commonly used in this area and is competitive with these other methods in terms of minimizing stress. These experiments were conducted using two eight-band images ($H = 8$) from the LANDSAT remote sensing database [1]. These include the New Zealand dataset shown in Figure 1 and the Tenerife dataset in Figure 6.

Accuracy: In the first set of experiments, we used 64×64 versions of these images ($N = 4096$) to allow computing ground truth solutions with full metric MDS and generated two-dimensional images ($L = 2$) as this was the maximum number the available implementation of Glimmer could support. Figure 5 plots the accuracy of each algorithm with respect to full MDS for an increasing number of landmarks K . We ran our algorithm using a range of subspace sizes ($M = 5, 10, 50, 100$). The plots give the *Procrustes Statistic* [11], measuring the minimum Euclidean distance between the output and the results of full MDS over all rotations, dilations and translations. A value of 0.0 indicates an optimal result and 1.0 corresponds to random noise.

As the plots indicate, PCA and Pivot MDS (for $K > 8$) are both close to optimal whereas our approach requires a larger K to achieve comparable accuracy. Additionally, we see that force-based methods such as Glimmer are not as accurate.

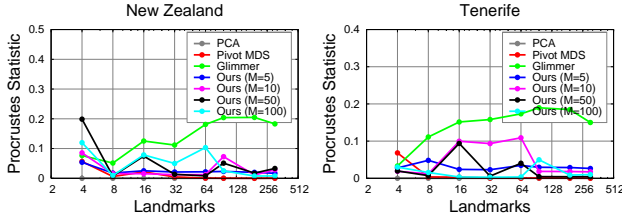


Fig. 5. Accuracy of our algorithm compared to SVD, Pivot MDS and Glimmer.

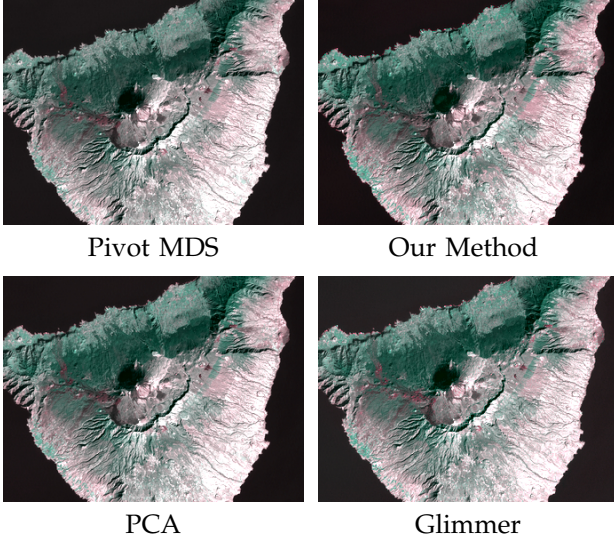


Fig. 6. Images produced by all four techniques for the Tenerife dataset (cropped). The output of each method was rigidly aligned to the PCA result to aid side-by-side comparison. The visual differences are minor.

To provide a visual interpretation of these errors, Figure 6 shows the results for the Tenerife dataset at 512×512 spatial resolution, obtained using each algorithm.⁴ We normalized each of the visualizations by applying the rigid body transformation that best aligns it with the PCA result. Note that despite the differences in Procrustes distances, Figure 6 indicates that all four methods produce results that are visually similar.

Efficiency: We also measured the performance of these four algorithms as a function of dataset size. The graphs along the top row in Figure 7 plot running time measured in seconds as a function of N and the graphs along the bottom plot the number of iterations required for convergence for Glimmer and our method. These graphs show results for images of sizes up to 512×512 . We used MATLAB's built-in "economy" SVD and we implemented Pivot MDS in MATLAB using optimized matrix functions wherever possible. All measurements were made using a Dell XPS with two Intel Core2 Quad CPUs and 3GB of

4. Based on our findings in Figure 5, we chose values of $M = 50$ and $K = 160$ for our technique and $K = 16$ for Pivot MDS. We used $K = 8$ for Glimmer as reported in their paper [46].

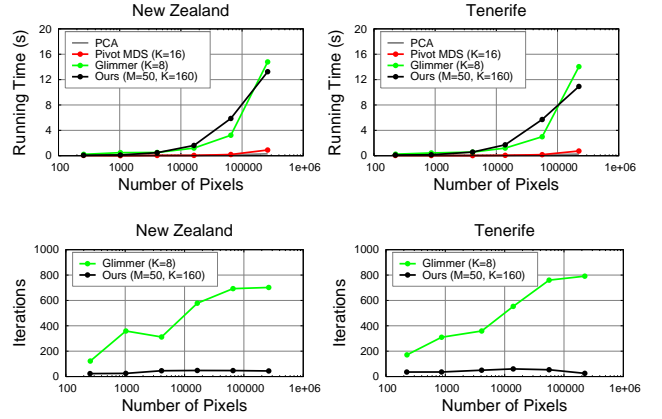


Fig. 7. Speed of our algorithm compared to PCA, Pivot MDS and Glimmer.

memory and an NVIDIA GeForce GTX 280.

Pivot MDS and PCA are efficient techniques and outperformed both Glimmer and our algorithm; note that the size of the respective covariance matrices are K^2 and H^2 . Note that Glimmer and our method are comparable in terms of speed. On the one hand, our approach requires a nearly constant number of iterations due to the fact that it is computing a fixed number of ML coordinates irrespective of the input size, whereas the number of iterations in Glimmer is roughly linear in the input size (Figure 7, bottom). On the other, the operations performed in each iteration of a force-directed method are cheaper and more parallelizable, leading to comparable running times.

5.2 Incorporating Constraints and non-Euclidean Distances

The experiments from the previous section indicate that PCA and Pivot MDS both provide comparably high accuracy at lower computational cost when minimizing raw stress. We now consider the qualitative advantages of solving the more general formulation of stress, which is not supported by either of these methods.

For these experiments, we have focused on 3-dimensional output color images, interpreting the output S as coordinates in the CIELAB color space. This helps ensure that distances between input pixel values are preserved as perceptual distances in the colors used to visualize them [47]. Additionally, we achieve a more intuitive range for w_i (pixel value constraints) by multiplying this value by N (the total number of pixels). This balances the contribution of the distance and value constraints in Equation 1, so that weights $w_{i,j} = w_i = 1.0$ give roughly equal weight to the two goals of stress minimization.

Color and Intensity Constraints: A number of researchers have discussed the importance of selecting color and intensity schemes appropriate for scientific visualizations [6], [48]. Although a complete analysis

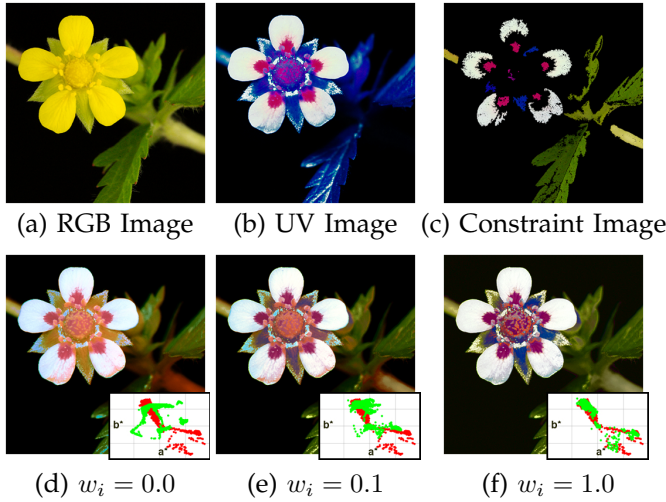


Fig. 8. Chromatic constraints applied to a 6-D multi-spectral image of a *Potentilla* flower. The constraints were constructed from the visible and UV images. The bottom images were computed by our algorithm for increasing values of w_i and include scatter plots showing the distribution of a^* , b^* coordinates in the constraint image (red) and for each of our results (green). Images © Bjørn Rørslett/NN/Nærfoto www.naturfotograf.com.

of the benefits of different palettes is beyond the scope of this work, we show that our technique can accommodate a wide range of useful constraints. In particular, we consider applications in which constraints are specified on either the chromatic bands or the intensity band. (The separability of the derivation in Appendix A allows us to specify weights on the different bands independently.)

Figure 8 shows an example of applying color constraints to a six-band multispectral image of a *Potentilla* flower. The input consists of a standard visible (RGB) image and one captured under ultra-violet (UV) light. These types of datasets help biologists study the way plants are perceived by different organisms. We created a set of chromaticity constraints that combine the green colors of the stems and leaves in the RGB image with the more vibrant colors of the petals and stamen in the UV image (Figure 8c). Note that these constraints are sparse, they were generated by simply copying regions of these two images into a common buffer. Images computed by our algorithm are shown along the bottom for increasing values of w_i (pixels without color constraints have $w_i = 0$). We initialized our algorithm using the constraint image. The accompanying graphs plot the a^* , b^* coordinates of the constraints (red) and our results (green). For a value of $w_i = 0.0$, the algorithm’s choice of colors is unconstrained and it is purely driven by the goal of minimizing stress. For increasing values of $w_i = 0.1$ and $w_i = 1.0$, the target colors are more faithfully captured as seen in both the images and the scatter

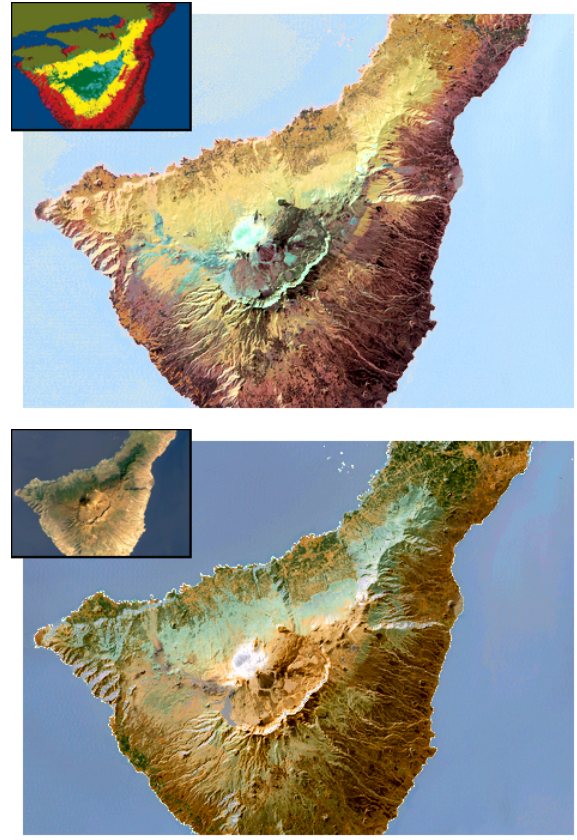


Fig. 9. Chromatic constraints applied to an 8-D multi-spectral aerial image of Tenerife, Spain. **Top:** Image produced by our algorithm using constraints derived from an existing false-color visualization of land cover [49]. **Bottom:** Alternative result produced by our algorithm that uses color constraints derived from the visible bands.

plots.

Figure 9 shows another example of chromatic constraints applied to a multispectral aerial image of Tenerife, Spain [1]. This dataset contains eight bands that sample the visible range (RGB) and near to far regions of the infrared (IR). The result at the top uses color constraints derived from a false-color visualization of this island’s land cover produced by Keuchel et al. [49], shown in the inset. Note that our result maintains the same basic palette of dark reds, yellows, greens and blues. However, unlike the Keuchel visualization, which was computed by quantizing the hyperspectral data into a custom color palette, our result reveals far greater detail and accurately conveys the relationships *between* these different regions as smooth color transitions that are proportional to the variation in the underlying data.

The image at the bottom of Figure 9 shows the result of using color constraints derived from the visible bands, shown in the inset. Specifically, the a^* , b^* coordinates in the visible image were translated to have zero mean and scaled by a factor of 1.5 to

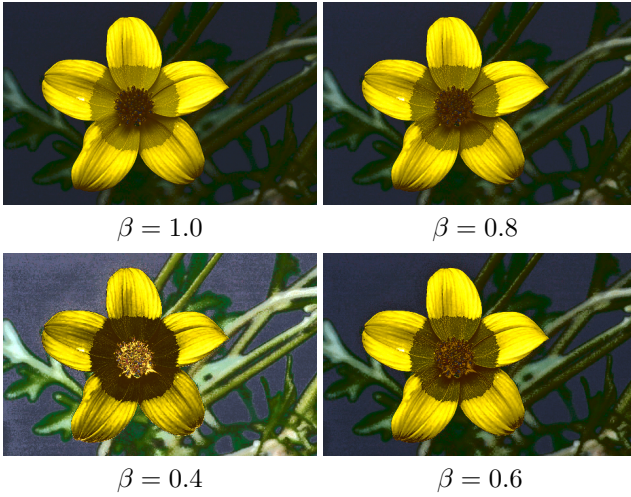


Fig. 10. Compressing distances. Images computed with our algorithm for decreasing values of β (clockwise from top left). Lower values more aggressively compress large distances, giving a flatter and more stylized appearance. Images © Bjørn Rørslett/NN/Nærfoto www.naturfotograf.com.

produce more vibrant, saturated colors. We found that these settings produced a more stylized visualization reminiscent of a hand-drawn illustration. This same approach was used in Figures 1 and 13.

Figure 12 shows an example of intensity constraints. This dataset consists of four images of the Nebula NGC 6611, also called the Eagle Nebula, taken by the Hubble Wide-Field Planetary Camera 2.⁵ In this case, it's important that stars and free space remain bright and dark, respectively. To accommodate this, we set intensity constraints equal to the norm of the pixel values in the input (see supplemental) and apply a low weight of $w_i = 0.01$; we initialized our algorithm using an RGB image constructed from the first three bands.

Exaggerating Details: Important features in a dataset often occur at different scales. For example, the variations *within* the ocean water and terrain, respectively, in Figure 1 are minor compared to the variations *between* these two regions. In order to capture both large- and small-scale features in a single visualization we modify the target pairwise distances $\delta_{ij}^{\mathbf{R}}$ in a way inspired by the tone-mapping operator of Fattal et al. [38]. Specifically, we set the target distances $\delta_{ij}^{\mathbf{R}}$ in Equation 1 to be the modified Euclidean distances

$$\delta_{ij}^{\mathbf{R}} = \|\mathbf{r}_i - \mathbf{r}_j\| \left(\frac{\alpha}{\|\mathbf{r}_i - \mathbf{r}_j\|} \right)^{1-\beta}. \quad (6)$$

The parameter β determines the degree of compression (for $\beta < 1.0$) and α defines a cutoff value below which distances are slightly amplified and above which they are compressed. Their effect is illustrated

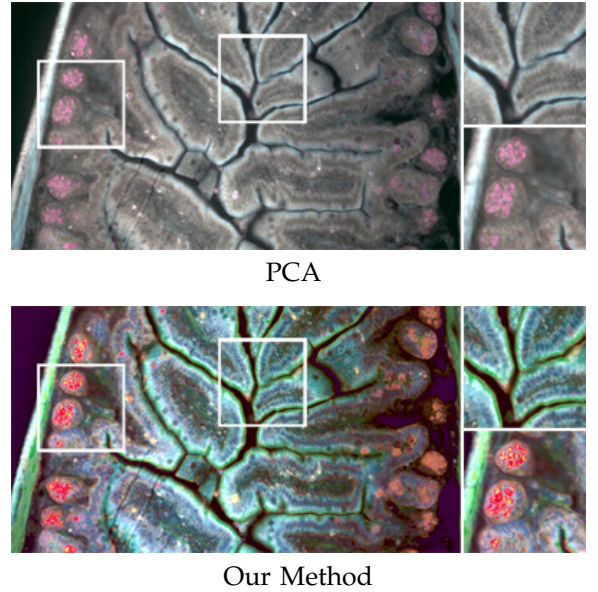


Fig. 11. Visualizations of a 31-D hyperspectral image of a mouse intestine. Our algorithm reveals more details than an image computed using PCA due to the fact that we use transformed distances ($\alpha = 0.1$ and $\beta = 0.6$). Multispectral data courtesy of Cambridge Research & Instrumentation, Inc.

in Figure 10, which shows several renditions of a 4-D multispectral image of a *Coreopsis* flower for a fixed value of $\alpha = 0.2$ and varying values of β . (The original RGB+UV images can be seen in Figure 2.) Lower values of β emphasize the differences between the petals and stamen and between the foreground and background. We softly constrained the chromatic bands to match the RGB image ($w_i = 0.01$) and also used these bands to initialize our algorithm. We found that small values of β produce flatter, more stylized images. Although the choice of α depends on the dataset and the desired effect, our implementation uses normalized distances and we found that in practice, setting α to be in the range $[0.1, 0.4]$ provided good results.

Figure 11 shows another example of exaggerating details in a multidimensional image, comparing

Dataset	w_i	α	β	Size	Time
New Zealand (Fig. 1)	0.1	0.15	0.3	1	59
Potentilla (Fig. 8)	-	0.2	0.8	0.5	21
Tenerife (Fig. 9, top)	0.1	0.2	0.8	0.5	14
Tenerife (Fig. 9, bottom)	0.1	0.2	0.8	0.5	11
Coreopsis (Fig. 10)	0.01	0.2	-	0.2	2
Intestine (Fig. 11)	0.01	0.1	(0.6)	0.2	34
Nebula (Fig. 12)	0.01	0.4	0.5	0.5	29
Grand Canyon (Fig. 13)	0.1	0.1	0.3	1	40

TABLE 1

Parameter settings for the results reported, along with the spatial resolution in megapixels (MP), and total running time of our algorithm in seconds.

5. <http://archive.eso.org/cms>

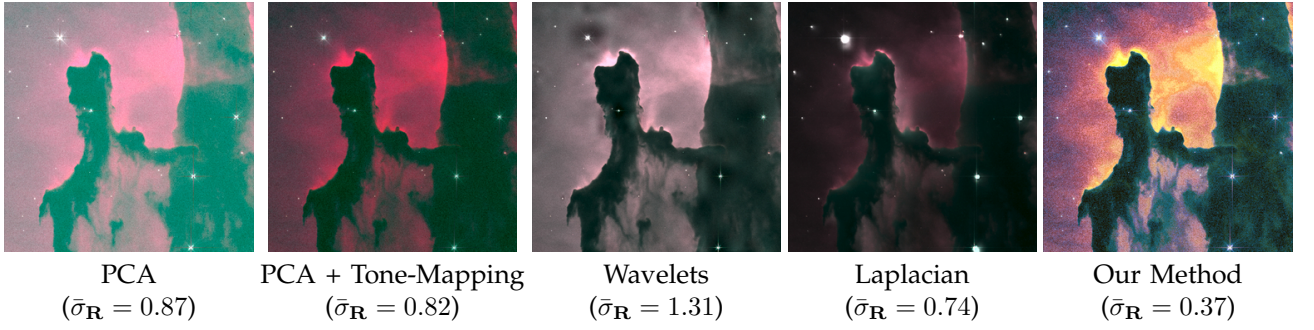


Fig. 12. Visualizations of a 4-D image of the Eagle Nebula computed using different image fusion techniques.

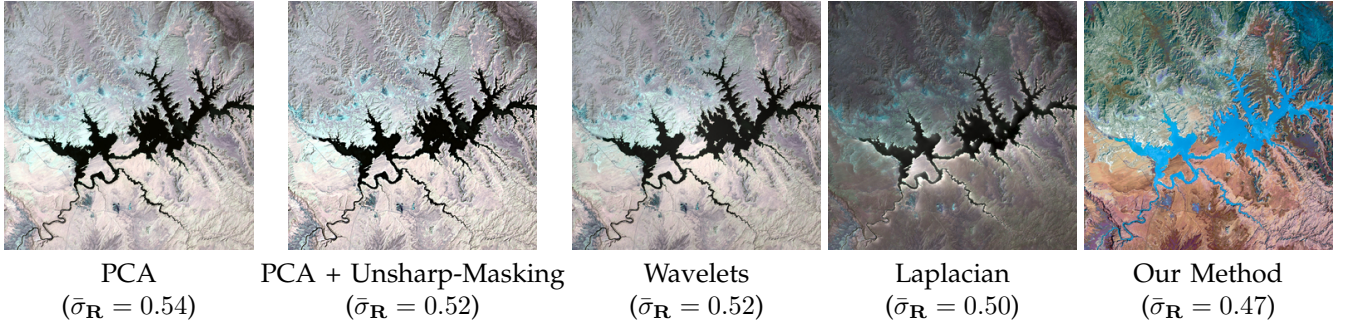


Fig. 13. Visualizations of an 8-D aerial image of the Grand Canyon [1] computed using different image fusion techniques.

the result of our method to the one obtained using PCA. This dataset contains 31 bands sampled at 10-nm intervals over the visible portion of the EM spectrum captured using a Nuance© Multispectral Imaging System mounted on an epifluorescence microscope. Note that the visualization computed with our method reveals fine-scale details in the tissue not easily visible in the PCA result.

The parameters used to create all of the results in this paper, along with the dataset sizes and total running times, are listed in Table 1. Almost all of the results in this paper show some amount of detail exaggeration. (This was particularly necessary for the remote sensing datasets in Figures 1 and 13 and the astronomy dataset in Figures 12, which have a dynamic range of 1000:1.)

5.3 Measuring Visualization Fidelity

As demonstrated in Figure 11, the ability to incorporate constraints into the optimization can result in visualizations that are richer than those derived by only minimizing raw stress. However, a natural question to consider is whether the obtained visualizations are “true” to the original input. To answer this question, we propose a simple measure of visualization fidelity. Using this metric, we compare our results to those of several common visualization techniques.

Defining a Fidelity Metric: The design of our fidelity metric is motivated by the observation that a good visualization should allow a viewer to infer relationships between pixels in the input by only considering

the values of pixels in the output image. As a simple measure, we would like to determine how often pixel similarity in the output visualization corresponds to pixel similarity in the input data.

To this end, we use a divisive clustering algorithm to separate an output image (\mathbf{S}) into a sufficient number of clusters so that they all have the same *intra-cluster* variance $\sigma_{\mathbf{S}}$. Then, we compute the average variance $\bar{\sigma}_{\mathbf{R}}$ of the input (\mathbf{R}) within each of these clusters and normalize by the total variance of the input. Smaller values of $\bar{\sigma}_{\mathbf{R}}$ indicate that uniform regions in the output visualization cover a region of the input whose values are also (nearly) uniform, corresponding to more faithful visualizations, while larger values correspond to lower fidelity visualizations.

Comparing to Existing Techniques: We compare the fidelity of our results to three state-of-the-art methods:

- 1) PCA: The high-dimensional input is projected onto the 3-D subspace defined by its directions of maximum variance and these values are interpreted as CIELAB coordinates. We also show the result of applying a tone-mapping filter [39] and an unsharp-masking filter as a post-process.
- 2) Wavelet-Based Fusion: A single intensity image in the wavelet domain is reconstructed by recording the maximum coefficient at each pixel across all of the bands [9]. A bi-orthogonal wavelet basis was used as this gives the fewest artifacts and produces results indistinguishable from state-of-the-art wavelet fusion meth-

ods [50]. Since this approach only provides a single-channel output, we use this output to set the luminance channel of the CIELAB image and transfer the a^* and b^* coordinates from PCA to the results, a common practice [10].

- 3) Laplacian Pyramids: Similar to wavelet-based fusion, a single intensity image is reconstructed by recording the maximum response at each pixel at each scale across the Laplacian Pyramids [8] of the component image. Again, we transferred the a^* and b^* coordinates from PCA to the results.

Figures 12 and 13 show the output of these techniques for a 4-D image of the Eagle Nebula and an 8-D image of the Grand Canyon [1]. For each visualization, the figures provide the average variance $\bar{\sigma}_R$ computed using a variance of $\sigma_S = 5.0$ for performing the clustering of output pixels values. These same trends were observed for different variance values $\sigma_S \in [2, 15]$. The fact that our method is uniformly lower supports the claim that it produces accurate visualizations that emphasize the input's structure.

Additionally, it is interesting to note that though the use of tone-mapping and unsharp-masking improves the fidelity of the PCA visualization, the resulting visualizations still have large $\bar{\sigma}_R$ values. This suggests that the non-linear exaggeration of detail should be incorporated into the optimization process and is less effective when used as a post-processing step.

6 LIMITATIONS AND FUTURE WORK

One limitation of our algorithm is that its output is influenced by its starting position. Although it produces an accurate visualization (low stress) from a random starting configuration, it is almost always desirable to guide this process by specifying color and/or intensity constraints and a useful initialization. While we see this flexibility as an advantage over purely automatic methods, it does require some amount of user effort and randomly seeded results must often be further processed by rotating, translating or inverting the computed colors. We did not perform this type of post-processing to the results in this paper to make our comparisons to fully automatic methods more fair.

Since our algorithm considers unsigned distances, it does not guarantee that spatial gradients in the input are preserved in the output. We observed this behavior for datasets with a very high dynamic range such as the astronomy examples. However, we found that applying soft constraints to the intensity channel was a reliable way to avoid these artifacts.

Improving the speed of our algorithm is another area of future work. The fact that our technique converges within a fixed number of iterations suggests a focus on reducing the cost of each iteration. Developing approximate or incremental strategies for updating these Laplacian matrices is one possible direction of study.

7 CONCLUSION

We have presented a new image fusion method based on Multidimensional Scaling. The key advantages of our approach over prior work are that it provides a principled way of accommodating user constraints on the computed visualization and allows a novel way of exaggerating details in the input. This was accomplished through an algorithm inspired by techniques developed in the graph drawing community that conducts the underlying optimization in a reduced subspace. We reported results for a variety of datasets drawn from remote sensing, biology and astronomy.

APPENDIX

DERIVATION OF MAJORIZATION UPDATE RULES

Here we derive the update rules for majorizing the stress:

$$\tau(\mathbf{S}) = \sum_{i < j} w_{ij} (\delta_{ij}^{\mathbf{R}} - \|\mathbf{s}_i - \mathbf{s}_j\|)^2 + \sum_i w_i \|\mathbf{s}_i - \bar{\mathbf{s}}_i\|^2.$$

Following the approach of Borg *et al.* [42], we expand the stress as the sum of three terms:

$$\tau(\mathbf{S}) = \eta_\delta^2 + \eta^2(\mathbf{S}) + 2\rho(\mathbf{S}),$$

where the three summands are defined as:

$$\begin{aligned} \eta_\delta^2 &= \sum_{i < j} w_{ij} (\delta_{ij}^{\mathbf{R}})^2 \\ \eta^2(\mathbf{S}) &= \sum_{i < j} w_{ij} \|\mathbf{s}_i - \mathbf{s}_j\|^2 + \sum_{1 < i \leq n} w_i \|\mathbf{s}_i - \bar{\mathbf{s}}_i\|^2 \\ \rho(\mathbf{S}) &= - \sum_{i < j} w_{ij} \delta_{ij}^{\mathbf{R}} \|\mathbf{s}_i - \mathbf{s}_j\|. \end{aligned}$$

and $\{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ denote the column vectors of \mathbf{S} . In this expression, the first term, η_δ^2 , is a constant, and does not play into the optimization. The second term, $\eta^2(\mathbf{S})$, is quadratic, positive, semi-definite, in \mathbf{S} , and can therefore be solved directly. It is the third term, $\rho(\mathbf{S})$, that makes minimizing the stress difficult.

To address this challenge, Borg *et al.* show that for any \mathbf{Z} , one can construct a linear function $\rho_{\mathbf{Z}}(\mathbf{S})$ that bounds ρ from above and agrees with ρ at \mathbf{Z} :

$$\rho_{\mathbf{Z}}(\mathbf{S}) \geq \rho(\mathbf{S}) \quad \text{and} \quad \rho_{\mathbf{Z}}(\mathbf{Z}) = \rho(\mathbf{Z}).$$

To construct this function they use the Cauchy-Schwarz inequality (i.e. the statement that the inner-product of two vectors is no larger than the product of their lengths) to get:

$$\begin{aligned} \rho(\mathbf{S}) &= - \sum_{i < j} w_{ij} \delta_{ij}^{\mathbf{R}} \frac{\|\mathbf{s}_i - \mathbf{s}_j\| \cdot \|\mathbf{z}_i - \mathbf{z}_j\|}{\|\mathbf{z}_i - \mathbf{z}_j\|} \\ &\leq - \sum_{i < j} \frac{w_{ij} \delta_{ij}^{\mathbf{R}}}{\|\mathbf{z}_i - \mathbf{z}_j\|} \langle \mathbf{s}_i - \mathbf{s}_j, \mathbf{z}_i - \mathbf{z}_j \rangle \\ &\equiv \rho_{\mathbf{Z}}(\mathbf{S}). \end{aligned}$$

This expression for $\rho_{\mathbf{Z}}(\mathbf{S})$ is linear in \mathbf{S} , it bounds $\rho(\mathbf{S})$ from above, and it agrees with $\rho(\mathbf{S})$ at $\mathbf{S} = \mathbf{Z}$. Thus, by replacing the stress $\tau(\mathbf{S})$ with its approximation at \mathbf{Z} :

$$\tau_{\mathbf{Z}}(\mathbf{S}) = \eta_{\delta}^2 + \eta^2(\mathbf{S}) + 2\rho_{\mathbf{Z}}(\mathbf{S}),$$

one obtains a quadratic, positive, semi-definite energy that can be minimized by solving a linear system of equations. Furthermore, since this quadratic energy bounds the stress from above and since it agrees with the stress at \mathbf{Z} , it follows that if $\tilde{\mathbf{Z}}$ is the minimizer of the quadratic energy, then the stress at $\tilde{\mathbf{Z}}$ cannot be greater than the stress at \mathbf{Z} .

Expanding out the approximation for the stress at \mathbf{Z} gives:

$$\tau_{\mathbf{Z}}(\mathbf{S}) = \eta_{\delta}^2 + \text{Tr}(\mathbf{S}^T \mathbf{L} \mathbf{S}) + \text{Tr}(\hat{\mathbf{S}}^T \Lambda \hat{\mathbf{S}}) - 2\text{Tr}(\mathbf{S}^T \mathbf{L}^Z \mathbf{Z})$$

where $\hat{\mathbf{S}} = \mathbf{S} - \bar{\mathbf{S}}$ is the vector of offsets from the prescribed values, \mathbf{L} is a Laplacian matrix whose entries are defined by the distance weights, Λ is the diagonal matrix of constraint weights, and \mathbf{L}^Z is a Laplacian matrix whose entries adjust for the relationship between the prescribed target distances and the distances realized by \mathbf{Z} :

$$\mathbf{L}_{ij} = \begin{cases} -w_{ij} & \text{for } i \neq j \\ \sum_{l \neq i} w_{il} & \text{for } i = j \end{cases}, \quad \Lambda_{ij} = \begin{cases} w_i & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases},$$

$$\mathbf{L}_{ij}^Z = \begin{cases} -\frac{w_{ij} \delta_{ij}^R}{\|\mathbf{z}_i - \mathbf{z}_j\|} & \text{for } i \neq j \text{ and } \|\mathbf{z}_i - \mathbf{z}_j\| \neq 0 \\ 0 & \text{for } i \neq j \text{ and } \|\mathbf{z}_i - \mathbf{z}_j\| = 0 \\ -\sum_{l \neq i} \mathbf{L}_{il} & \text{for } i = j \end{cases}.$$

Thus, given an estimate for the minimizer of stress, $\mathbf{Z} = \mathbf{S}^k$, the next estimate, \mathbf{S}^{k+1} can be obtained by solving for the image \mathbf{S}^{k+1} that minimizes $\tau_{\mathbf{Z}}(\mathbf{S})$. Setting the gradient to zero, \mathbf{S}^{k+1} is obtained by solving the symmetric, positive, semi-definite system:

$$(\mathbf{L} + \Lambda) \mathbf{S}^{k+1} = \mathbf{L}^Z \mathbf{Z} + \Lambda \bar{\mathbf{S}}.$$

ACKNOWLEDGEMENTS

The authors wish to thank Bjørn Rørslett with /NN/Nærfoto www.naturfotograf.com for the hyperspectral images of flowers in Figures 2 and 8. We also thank Cambridge Research & Instrumentation, Inc. for providing the multispectral data used in Figure 11 which was captured using their Nuance© Multispectral Imaging System.

REFERENCES

- [1] Landsat, 2008. [Online]. Available: <http://landsat.gsfc.nasa.gov>
- [2] P. J. Burt and R. J. Kolczynski, "Enhanced image capture through fusion," in *Proceedings of Computer Vision*, 1993, pp. 173–182.
- [3] C. Pohl and J. L. V. Genderen, "Multisensor image fusion in remote sensing: Concepts, methods and applications," *International Journal of Remote Sensing*, vol. 19, no. 5, pp. 823–854, 1998.
- [4] R. S. Blum and Z. Liu, Eds., *Multi-sensor Image Fusion and its Applications*. Taylor & Francis, 2006.
- [5] 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*, vol. 41, no. 3, pp. 708–718, 2003.
- [6] N. P. Jacobson and M. R. Gupta, "Design goals and solutions for display of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 11, pp. 2684–2692, 2005.
- [7] P. J. Ready and P. A. Wintz, "Information extraction, SNR improvement, and data compression in multispectral imagery," *IEEE Transactions on Communications*, vol. 41, no. 3, pp. 1123–1131, 1973.
- [8] A. Toet, "Hierarchical image fusion," *Machine Vision and Applications*, vol. 3, no. 1, pp. 1–11, 1990.
- [9] H. Li, B. S. Manjunath, and S. K. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical Models and Image Processing*, vol. 57, no. 3, pp. 235–245, 1995.
- [10] D. A. Socolinsky and L. B. Wolff, "Multispectral image visualization through first-order fusion," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 923–931, 2002.
- [11] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd ed. Chapman & Hall/CRC, 2001.
- [12] D. Harel and Y. Koren, "Graph drawing by high-dimensional embedding," *Journal of Graph Algorithms and Applications*, vol. 8, no. 2, pp. 195–214, 2004.
- [13] Y. Koren, "Graph drawing by subspace optimization," in *Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization*, 2004, pp. 65–74.
- [14] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems (NIPS)*, 2003, pp. 705–712.
- [15] U. Brandes and C. Pich, "Eigensolver methods for progressive multidimensional scaling of large data," in *Proceedings of Graph Drawing*, 2006, pp. 42–53.
- [16] A. Ifarraguerri and C.-I. Chang, "Hyperspectral image segmentation with convex cones," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 756–770, 1999.
- [17] J. Silverman, C. E. Cafer, J. M. Mooney, M. M. Weeks, and P. Yip, "Automated clustering/segmentation of hyperspectral images based on histogram thresholding," in *Proceedings of SPIE*, vol. 4480, no. 65, 2002, pp. 65–75.
- [18] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. 23, no. 9, pp. 881–889, 1974.
- [19] G. Harikumar and Y. Bresler, "Feature extraction techniques for exploratory visualization of vector-valued imagery," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1324–1334, 1996.
- [20] X. Jia and J. A. Richards, "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 538–542, 1999.
- [21] Q. Du, S. Cai, and R. J. Moorhead, "Color display for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1858–1866, 2008.
- [22] M. Cui, A. Razdan, J. Hu, and P. Wonka, "Interactive hyperspectral image visualization using convex optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1673–1684, 2009.
- [23] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 1995.
- [24] A. Manduca, "Multispectral image visualization with nonlinear projections," *IEEE Transactions on Image Processing*, vol. 5, no. 10, pp. 1486–1490, 1996.
- [25] Z. Zhang, S. Sun, and F. Zheng, "Image fusion based on median filters and SOFM neural networks: A three-step scheme," *Signal Processing*, vol. 81, no. 6, pp. 1325–1330, 2001.
- [26] D. H. Kim and L. H. Finkel, "Hyperspectral image processing using locally linear embedding," in *Proceedings of the 1st International IEEE Conference on Neural Engineering*, 2003, pp. 316–319.
- [27] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

- [28] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch, "Color2Gray: Saliency-preserving color removal," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 24, no. 3, pp. 634–639, 2005.
- [29] K. Rasche, R. Geist, and J. Westall, "Re-coloring images for gamuts of lower dimension," *Computer Graphics Forum (Proc. of EUROGRAPHICS)*, vol. 24, no. 3, pp. 423–432, 2005.
- [30] K. Smith, J. T. Pierre-Edouard Landes, and K. Myszkowski, "Apparent greyscale: A simple and fast conversion to perceptually accurate images and video," *Computer Graphics Forum (Proc. of EUROGRAPHICS)*, vol. 27, no. 2, pp. 193–200, 2008.
- [31] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [32] P. Scheunders, "Multiscale edge representation applied to image fusion," in *Proceedings of SPIE*, 2000, pp. 894–901.
- [33] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, 1992.
- [34] S. D. Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 116–125, 1986.
- [35] D. A. Socolinsky and L. B. Wolff, "A new visualization paradigm for multispectral imagery and data fusion," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. 319–324.
- [36] J. Tumblin, "Three methods of detail-preserving contrast reduction for displayed images," Ph.D. dissertation, Georgia Institute of Technology, 1999.
- [37] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proceedings of ACM SIGGRAPH*, 2002, pp. 257–266.
- [38] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *Proceedings of ACM SIGGRAPH*, 2002, pp. 249–256.
- [39] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," in *Proceedings of ACM SIGGRAPH*, 2002, pp. 267–276.
- [40] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007.
- [41] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 27, no. 3, 2008.
- [42] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, 1997.
- [43] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [44] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. Barra, F. Brodeau, G. Romier, and B. V. Cutsem, Eds. Amsterdam: North Holland Publishing Company, 1977, pp. 133–146.
- [45] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Proceedings of the International Symposium on Graph Drawing*, 2004, pp. 239–250.
- [46] S. Ingram, T. Munzner, and M. Olano, "Glimmer: Multilevel MDS on the GPU," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 249–261, 2009.
- [47] M. D. Fairchild, *Color Appearance Models*, 1st ed. Addison Wesley, 1997.
- [48] E. R. Tufte, *Visual Explanations: Images and Quantities, Evidence and Narrative*, 1st ed. Graphics Press, 1997.
- [49] J. Keuchel, S. Naumann, M. Heiler, and A. Siegmund, "Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data," *Remote Sensing of Environment*, vol. 86, no. 4, pp. 530–541, 2003.
- [50] J. J. Lewis, R. J. O'Callaghan, S. G. Nikolov, D. R. Bull, and N. Canagarajah, "Pixel- and region-based image fusion using complex wavelets," *Image Fusion*, vol. 8, no. 2, pp. 119–130, 2007.



Jason Lawrence received a PhD degree in computer science from Princeton University in 2006. He is currently an assistant professor in the Department of Computer Science at the University of Virginia. His research interests include shape and material acquisition, representations of material appearance, and interactive and global illumination rendering algorithms.



Sean Arietta received a BS degree in physics from the University of Virginia in 2008. He is currently a PhD student at the University of Virginia in the Department of Computer Science. His research interests include image enhancement and restoration, texture synthesis, large-scale image processing, and content-based image retrieval.



Michael Kazhdan received a PhD degree in computer science in 2004 from Princeton University. He is currently an assistant professor in the Computer Science Department of Johns Hopkins University. His research interests include shape analysis, surface reconstruction, and large image processing.



Daniel Lepage received BA degrees in computer science and mathematics from Cornell University. He is currently a PhD candidate in computer science at the University of Virginia. His research interests include material representations and tools for editing complex visual datasets.



Colleen O'Hagan received a BS degree in electrical engineering from the University of Notre Dame in 2007 and the MS degree in computer science from the University of Virginia in 2009. She is currently working in the film industry on visual effects and computer animation.