

Multi-Channel Transfer Function with Dimensionality Reduction

Han Suk Kim^{a, c}, Jürgen P. Schulze^b, Angela C. Cone^c,
Gina E. Sosinsky^c, and Maryann E. Martone^c

^aDepartment of Computer Science and Engineering

^bCalifornia Institute for Telecommunications and Information Technology

^cNational Center for Microscopy and Imaging Research

University of California San Diego, 9500 Gilman Drive, La Jolla, CA, USA

ABSTRACT

The design of transfer functions for volume rendering is a difficult task. This is particularly true for multi-channel data sets, where multiple data values exist for each voxel. In this paper, we propose a new method for transfer function design. Our new method provides a framework to combine multiple approaches and pushes the boundary of gradient-based transfer functions to multiple channels, while still keeping the dimensionality of transfer functions to a manageable level, i.e., a maximum of three dimensions, which can be displayed visually in a straightforward way. Our approach utilizes channel intensity, gradient, curvature and texture properties of each voxel. The high-dimensional data of the domain is reduced by applying recently developed nonlinear dimensionality reduction algorithms. In this paper, we used Isomap as well as a traditional algorithm, Principle Component Analysis (PCA). Our results show that these dimensionality reduction algorithms significantly improve the transfer function design process without compromising visualization accuracy. In this publication we report on the impact of the dimensionality reduction algorithms on transfer function design for confocal microscopy data.

Keywords: Volume Rendering, Transfer Function, Multi-Channel Volume, Light Microscopy Imaging, Dimensionality Reduction

1. INTRODUCTION

The transfer function is a key component of volume rendering systems. It maps data value to color (red, green, blue components) and opacity (alpha). Defining a meaningful mapping from measurement intensity or other variables to optical properties, however, is a difficult task. This is mostly because the volume data set often does not directly contain the kind of information one would need to create a good mapping. But it becomes easier if the transfer function design is started with a domain that distributes voxels into multiple clusters in the domain where each cluster represents a certain feature of the data, e.g., a boundary or the inside of objects in the data, for instance a cell structure. Feature isolation, or segmentation, is an essential part of transfer functions. One goal of transfer function design is to emphasize areas of interest with distinct colors.

In order to isolate features from other parts, mathematical properties of the data, e.g., gradient or curvature, have to be considered for the design of the transfer function. However, it is often not clear whether or not a property included in the domain will indeed help isolate features. Building a good starting domain gets more difficult when dealing with multi-channel data. Multiple intensities from multi-modal measurements already require a higher dimensional domain space. If one would like to consider gradient information, and even if the gradient information is encoded by gradient magnitude, the dimensionality of the domain is doubled. In this publication, we propose a framework that can potentially handle any number of features. The number of features is less restricted in our proposed method and users are free to add additional fields to the transfer

Further author information: (Send correspondence to Han Suk Kim)

Han Suk Kim: hskim@cs.ucsd.edu, Jürgen P. Schulze: jschulze@soe.ucsd.edu, Angela C. Cone: acone@ncmir.ucsd.edu, Gina E. Sosinsky: gsosinsky@ucsd.edu, and Maryann E. Martone: mmartone@ucsd.edu.

function domain. Once the domain has enough information to isolate areas of interest in the volume, the high dimensionality is reduced to the usual up to three dimensions of color (red, green, blue) by a state-of-the-art dimensionality reduction algorithm. The reduction algorithm preserves relative distances between points in high-dimensional space. Therefore, two points close to each other in the domain after dimensionality reduction also are close to each other in the original domain. The distance-preserving characteristics of dimensional reduction algorithms are crucial in the transfer function domain, because the concept of transfer function design is based on the assumption that points close to each other in the data domain share the same or similar properties and that if those points are assigned to the same or similar colors, we will perceive them as part of the same objects. While the dimensionality reduction algorithms automatically generate a domain, the freedom of visualizing the resulting lower dimensional domain stays the same as in traditional multi-channel transfer function domains. One can examine the domain to find areas of interest using any type of widgets to map color and opacity to the data values. The contributions of our new approach can be summarized as follows:

Multiple Features We propose a new framework in which multiple features of multiple channels can be integrated in the transfer function domain. Traditional approaches have restricted the domain to 2D or 3D because otherwise it is difficult to navigate it. There have also been many ways to avoid high dimensionality, such as data fusion,¹ or multiple levels of transfer function construction.² With our method, any number of features can be added to the domain, which gives visualization scientists a potential to easily find areas of interest in volume data.

Multiple Channels Our new method opens up a new way for visualization scientists, especially those who would like to visualize multi-channel data. Our method enables visualization scientists to examine data where two or more features have positive or negative correlations. Many scientific data sets have multi-variate information. Examining multiple fields at the same time, instead of visualizing one field at a time, allows visualization scientists to work more efficiently.

Dimensionality Reduction Recent research on non-linear dimensionality reduction algorithms has shown that the proposed algorithms can successfully find a low dimensional representation of very high-dimensional data. We apply this idea to the transfer function domain. We consider our work to be an example showing the effectiveness of dimensionality reduction algorithms in a field they have not previously been applied to.

This paper is organized as follows: Section 2 discusses previous approaches on transfer function design and dimensionality reduction algorithms. Then, Section 3 reviews the two dimensionality reduction algorithms we employed for our work. Section 4 presents how we apply dimensionality reduction algorithms to automate transfer function design for multi-channel data sets. Section 5 shows results from dimensionality reductions and volume rendering, followed by Section 6 discussing the results. Finally, Section 7 describes future work and concludes this paper.

2. RELATED WORK

Our new method combines algorithms from two different disciplines: multi-dimensional transfer functions from volume rendering and dimensionality reduction algorithms from machine learning. In the following, we briefly discuss previous work related to our approach.

2.1 Transfer Function

Due to the significance and the difficulty of finding a good transfer function in direct volume rendering, transfer functions have been an active area of research in the field of volume rendering.^{3,4} Especially, significant effort was made to extend the domain of transfer functions⁵ from 1D intensity-based transfer functions to multi-dimensional ones. The most popular choice for the second dimension has been gradient magnitude.^{5,6} Gradient magnitude encodes the boundaries of the data in the volume. A large value of gradient magnitude at a voxel indicates that the intensity changes rapidly around the voxel, whereas in smooth areas, gradient magnitude values are small. Since gradient magnitude as a second transfer function dimension has worked so well, many groups have proposed other derived values as the second dimension.

Curvature information^{7,8} was shown to be useful in non-photorealistic rendering by emphasizing ridges and valleys. Correa et al.⁹ considered relative sizes of features as an alternative domain. Caban et al.² employed texture properties to encode local textural information. First order statistics, such as mean, variance, and skewness, as well as second order statistics and run length matrices, were used to describe texture properties. This approach is related to our work in that the dimensionality of information reaches up to more than 20. The high dimensional descriptor enhances volume rendering by differentiating two parts that have similar intensity and gradient values but different textural characteristics. The high-dimensional information is used as secondary information to map each point to several classes grouped by textural similarities.

Human perception of 4D or higher dimensional data is very limited, and thus keeping the dimensionality of the domain to 2D or 3D is critical for the user interface. Therefore, all the above approaches limit the maximum dimensionality of the transfer function. However, our work aims to give a way to utilize multiple quantities all together in transfer function design.

2.2 Dimensionality Reduction

Besides the dimensionality reduction algorithms discussed in Section 3, many alternative algorithms have been proposed. Although PCA has been successfully used for a long time in many fields, recent publications in non-linear dimensionality reduction algorithms have shown great success in analyzing complex high-dimensional data.¹⁰⁻¹⁴ Each approach attempts to preserve different metrics and constructs different optimization problems under different constraints. However, the problem they all try to solve is the same, finding a low dimensional representation that faithfully describes the high-dimensional data.

In scientific visualization, self-organizing maps¹⁵ and K-means clustering¹⁶ have widely been used for data exploration. Although each algorithm has shown strength in data analysis to display the distribution of high-dimensional data, the results are subject to a set of initial reference vectors or initial K-means that users initially assign. The non-linear dimensionality reduction algorithms, on the other hand, require a minimum number of free parameters, e.g. K in the K-nearest neighborhood algorithm, and they find a low-dimensional representation that preserves relative distances in high-dimensional space. Moreover, non-linear dimensionality reduction algorithms produce a single globally optimal point, as opposed to many local minima in self-organizing map or in K-means clustering.

Lewis et al.¹⁷ applied K-means clustering, PCA, Isomap, and Maximum Variance Unfolding^{10,11} to oceanographic multivariate data in an approach similar to ours. They use clustering and classification of data into multiple disjoint groups, which is parallel to our work in that we focus on color encoding for transfer function design and feature discovery in volume visualization to emphasize areas of interest in biological images.

3. DIMENSIONALITY REDUCTION

In this section, we briefly review the two dimensionality reduction algorithms that we employ, namely, 1) Principle Component Analysis (PCA) and 2) Isomap. We discuss these algorithms in terms of what each of the algorithms preserves in the low-dimensional representation and what they minimize/maximize. Intuitively, the inputs to dimensionality reduction algorithms are a set of vectors in $D > 3$ dimensional space and the outputs are the same number of points but in 3D space. Preserved between the inputs and outputs are the distances between points. Readers are encouraged to see Ref. 18–21 for more details.

Let X_1, \dots, X_N be inputs in high dimensional space, \mathbb{R}^D . The goal of dimensionality reduction algorithms is to find a low dimensional representation $Y_1, \dots, Y_N \in \mathbb{R}^d$ such that $d \ll D$ and $\{Y_i\}_{i=1}^N$ faithfully explains the original data $\{X_i\}_{i=1}^N$.

3.1 Principle Component Analysis

The main idea of Principle Component Analysis (PCA) is to find a subspace that maximizes the variance when $\{X_i\}_{i=1}^N$ are projected to the subspace. For example, if X_i 's are spread out on or near the x-axis in \mathbb{R}^2 , a good 1D subspace would be the x-axis, rather than the y-axis: all the points projected to the y-axis are around $y = 0$ whereas projecting to the x-axis loses less information by preserving the variance in 2D. Therefore, PCA maximizes the variance of projected sample points to direction \hat{u} :

$$\hat{u} = \max_{\vec{u}} \frac{1}{N} \sum_{i=1}^N (\vec{X}_i \cdot \vec{u})^2 \quad (1)$$

The solution for this optimization problem is computed by singular value decomposition (SVD).²¹ SVD results in the eigenvector with the largest eigenvalue of covariance matrix C of sample points X_i . Namely, for a given set of points $\{\vec{X}_i\}_{i=1}^N$, PCA computes the covariance matrix C of X_i 's and solves eigenvalue equations $C\vec{u} = \lambda\vec{u}$. Then the top d eigenvectors that yield the largest eigenvalues are the direction \hat{u}_j ($j = 1 \dots d$) which achieves maximum variance. The j th component of Y_i ($j = 1 \dots d$) can be easily computed with the inner product of X_i and \hat{u}_j , i.e. $Y_{ij} = X_i \cdot \hat{u}_j$.

Due to the simplicity of the algorithm, albeit computationally expensive, PCA has been widely used in many fields. Note also that its solution is the global minimum. However, the limitation is the assumption that the sample points are on or near a linear subspace.

3.2 Isomap

Isomap^{18,19} is an extension of multidimensional scaling (MDS)²⁰ for finding a low-dimensional representation of samples that have nonlinear shapes. The algorithm assumes that the observation points in high-dimensional space are a sample from an intrinsic lower dimensional manifold. The key idea here is that the geodesic distance between two points, X_i and X_j , not the Euclidean distance in \mathbb{R}^D , captures the shape of non-linear structures in high-dimensional space. One canonical example widely used is a Swiss roll shaped plane. A long strip shaped plane is rolled into a spiral. Geodesic distance is the distance between two points on the Swiss roll when traveling only on the plane. Then, MDS constructs a low-dimensional representation $\{Y_i\}_{i=1}^N$ where the distance between two points in Y is closest to the geodesic distance.

The algorithm has three steps. The first step is to construct a neighborhood graph based on Euclidean distance in the high-dimensional space. For each point X_i , the distance to other points, $\|X_i - X_j\|_2$ is computed, and the closest K points are picked. The distance is used as the weight between the two nodes, X_i and X_j , in the neighborhood graph, but if two nodes are not close to each other, i.e., they are not in the K -nearest neighborhood to each other, no edge exists. The second step is to compute the shortest path between any two nodes. The resulting information, shortest distances between nodes, encodes the geodesic distance. The distance approximately measures the distance on the manifold. For example, if we have points sampled from a sphere, the geodesic distance represents the shortest distance when traveling only on the surface of sphere. Thus, this graph captures the nonlinear manifold. The last step is running MDS on the graph to find a low-dimensional representation preserving the pairwise geodesic distances.

The computational cost is higher than PCA because of the first and second step. A naive implementation of the K -nearest neighborhood can go up to $O(N^2D)$. However, a sophisticated implementation can optimize the process, and it runs much faster. Furthermore, the process is embarrassingly parallelizable, so OpenMP²² can be utilized to boost the performance on a multi-core shared memory system. The second step, with Dijkstra's algorithm²³ and its naive implementation, can run in $O(N^2)$, although it is known that a Fibonacci heap can reduce the asymptotic time down to $O(N \log N + E)$, where E is the number of edges in the graph.

The cost can be further reduced by using landmarks.¹⁹ Instead of using all the points in the analysis, which are redundant in most cases, only a small set of randomly chosen points, called landmarks, are considered. This algorithm dramatically reduces the computational cost while producing almost the same quality of the resulting low-dimensional representation.

4. TRANSFER FUNCTION DESIGN

Our goal in this work is to support the design of multi-dimensional transfer functions for multi-channel data sets. We define multi-channel data to be a set of sample data from the same object but with different types of measurement methods as described in Section 5.1. With multi-channel data, the areas scientists are most interested in, are often described in complex ways: for example, areas where first and third channels are active together and where the activation area is around the boundary of an object. This type of area cannot be found

with traditional 1D transfer functions because assigning color with regards to only the intensity of a single channel cannot capture the correlations between multiple channels.

The first step of our transfer function design process is collecting features of voxels. We construct feature vectors defined for each voxel to contain enough information to encode areas of interest. The feature vectors consist of 24 fields in our example. The first three fields are intensity information for each channel, (u_1, u_2, u_3) . The next information included is gradient information to isolate voxels near boundaries. Because we are not restricted to the size of feature vectors, which is one of the strengths of our approach, we add not only gradient magnitude, but also each component of the gradient vector, i.e. $\left(\frac{\partial u_i}{\partial x}, \frac{\partial u_i}{\partial y}, \frac{\partial u_i}{\partial z}\right)$ for $i = 1, 2, 3$. In addition, discrete Laplacian values are appended to capture ridges and valleys as in Ref., i.e., $\frac{1}{6} \left(\frac{\partial^2 u_i}{\partial x^2} + \frac{\partial^2 u_i}{\partial y^2} + \frac{\partial^2 u_i}{\partial z^2}\right)$ for $i = 1, 2, 3$. Finally, two values from first order statistics are added: mean and variance of channel intensity. In our example, 3x3x3 voxel cubes are used to capture textural characteristics.²

One may argue that the feature vector is superfluous, but even though it may have redundant information, the dimensionality reduction algorithms can filter out redundancies. Note also that one has all the freedom of adding or removing features; if a set of features is believed to be effective in isolation, adding to the current feature vector does not cause much overhead. If an added feature is new information, the dimensionality reduction algorithm will capture the large variance of the feature. Even if it turns out that they are not effective, the small variance of the domain will be eliminated also by the dimensionality reduction algorithm.

The second step is to run the dimensionality reduction algorithm. The input is the set of feature vectors we constructed in the first step. The reason why we apply the dimensionality reduction algorithm is that the dimensionality of the original feature vector is too high - 24 in our example but it may be even larger depending on the visualization goals. What we get after dimensionality reduction is a compact form of a voxel distribution in 3D. Instead of working with 24D data, the dimensionality reduction algorithm enables us to stay in 3D space. The relationship between the original 24D distribution and the newly computed 3D distribution is that the 3D distribution best approximates the original data. What is preserved when transforming feature vectors from 24D to 3D, is the relative distance between feature vectors. Two vectors which are close to each other in high-dimensional space are close in low-dimensional space as well. This allows us to assign similar colors to two voxels that are located close to each other in the resulting 3D domain because they are also close in the original domain.

The final step for our transfer function design is to navigate the domain produced in the second step. This is the same procedure visualization scientists go through to create multi-dimensional transfer functions in previous work. Dual domain navigation and coloring widgets⁵ help examining the domain in this step. The feature domain, which is actually produced by the dimensionality reduction algorithms, is created in such a way that large variance of fields in the original data are shown in the feature domain, too. Preserving the variance of the original data helps visualization scientists to more easily identify areas of interest in the volume rendering domain.

5. RESULT

We first introduce the volume data we used for our experiment in this section. Then we present 3D transfer function and direct volume rendering results, highlighting areas of interest, generated in the transfer function domain.

5.1 Light Microscopy Volume

Here we apply our new transfer function design process to a dataset of images acquired by confocal microscopy of immunolabeled mouse brain tissue. The tissue is labeled with 3 different fluorescent antibodies to localize three proteins within the tissue. The image consists of three channels, each of which is the emission of a different wavelength of light collected from these fluorophores within the specimen. This dataset was generated at National Center for Microscopy and Imaging Research (NCMIR) and deposited into the Cell Centered Database (CCDB).

The original microscopy product is a mosaic of 176 overlapping image stacks that were acquired and assembled into one image encompassing the entire hippocampus region of the mouse brain acquired using a laser scanning confocal microscope with a precision motorized stage. The original wide-field mosaic image can be browsed at

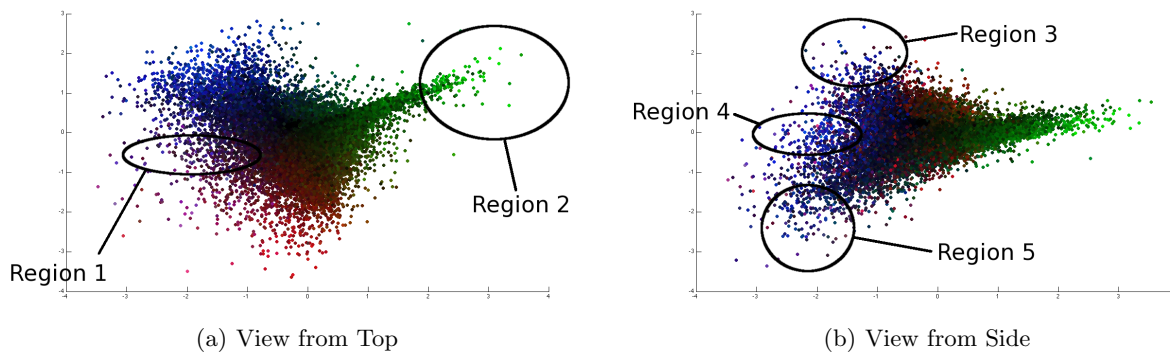


Figure 1. The domain distribution of feature vectors after Isomap dimensionality reduction. Figure 1(a) is the view from the top of the domain, i.e., projected to the XY plane. Figure 1(b) is the view from a side of the domain, i.e., projected to the XZ plane.

multiple resolutions of the entire specimen, regions of interest, certain cell populations within regions, and even cellular compartments (at maximum resolution). This large-scale imaging technique is used to scan large regions of the mouse brain to look for expression of Pannexin1 protein that has not previously been examined at the protein level in this tissue. In these images, Pannexin1 protein expression (green) is examined in comparison with that of Connexin43 protein (red) and the astrocyte marker, Glial Fibrillary Acidic protein (GFAP) (blue). The aim is to learn where Pannexin1 protein is being utilized in brain tissue at multiple levels of resolution and determine what other cellular partners might bind to the Pannexin1 protein assemblies.

This image data is rich in content, large in size and volumes like these usually contain much more information beyond the original experimental intent. They are good source material for data mining and sharing technology that stimulates new avenues of investigation. These large-scale mosaic image volumes must be cropped to a manageable size at an appropriate scale in order to demonstrate the findings for collaborators and in publications such as the examples in Figure 2.

5.2 Transfer Function Domain

Figure 1 shows the domain of a transfer function of the data described in Section 5.1. Figure 1(a) is the view from the top of the domain, i.e., from the positive infinite direction of the Z axis, or projection to the XY plane. Figure 1(b) is the view from one side of the domain, that is, from the positive infinite direction of the Y axis, or projection to the XZ plane. This is produced after applying the Isomap algorithm with 500 landmarks with $K = 50$ nearest neighborhood searchings. To give basic information about the distribution, the original channel intensities of the voxels were initially assigned to points.

In Figure 1(a), three channel intensity values are clustered at each corner. Voxels expressing two channels together are between two corners. Especially region 1 denotes where Astrocytes and Connexin43 proteins are expressed around the boundary of the cell. The distribution of voxels that express Pannexin1 protein branches out from the center, region 2. The edge of the branch is the voxels representing blobs of Pannexin1 as shown in Figure 2(c).

It is more clear to see how Connexin43 proteins are distributed in the sample with Figure 1(b). The voxels are spread according to ridges and valleys; region 3 is where valleys going down to $+z$ depth from right to left; region 4 is where ridge voxels are distributed; and region 5 denotes another valleys but the opposite direction to region 2.

Many widgets for editing transfer function, for instance as in Ref. 24, can be used in the same way with this domain. For the rendering images in Section 5.3, only a Gaussian widget was used.

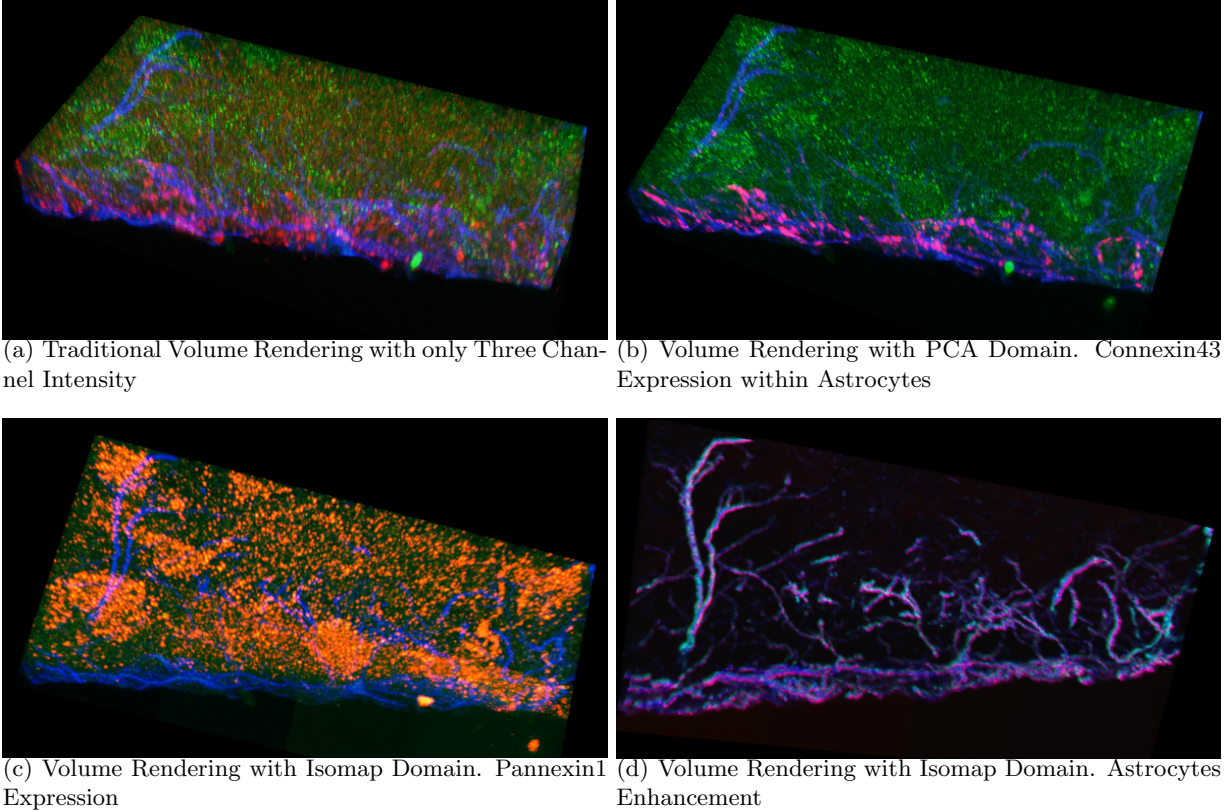


Figure 2. Rendering images of a three channel $512 \times 256 \times 24$ volume. All the images depict the same volume but with different transfer functions. Maximum Intensity Projection was used for rendering. Figure 2(a) is a result with a transfer function that maps each of three channel intensities to RGB color. Figure 2(b), 2(c), and 2(d) show examples of our methods. Each of the three images was generated with the domain that our dimensionality reduction algorithms were applied. Figure 2(b) shows the area where Connexin43 and Astrocytes are both expressed at the boundary. Figure 2(c) emphasizes where only Pannexin1 is expressed. Figure 2(d) shows where Astrocytes with valleys are highlighted with different colors.

5.3 Rendering Results

We rendered a small subset ($512 \times 256 \times 24$ voxels) with DeskVOX,²⁵ an open-source volume rendering software. A maximum intensity projection algorithm was used for blending the slices. Figure 2 shows several screen shots.

In Figure 2(a), a small region of the three-color image shows the edge of the hippocampus tissue with triple fluorescent immunolabeling (Green=Pannexin1 protein, Red=Connexin43 protein, Blue=GFAP within astrocytes). In Figure 2(a), a projection of the original multi-channel RGB image stack is displayed for comparison with improved visualizations of key elements generated using our new transfer function design process that are shown in Figure 2(b)-2(d).

The Astrocyte cells (blue in Figure 2) cover and wrap the outer surface of the hippocampus. These Astrocyte cells, as well as other cell types, express the gap junction protein Connexin43 (red). Figure 2(b) nicely highlights the Connexin43 (red) expression within astrocytes (blue) surrounding the hippocampus tissue while suppressing the Connexin43 labeling that is present among unlabeled, and therefore unidentifiable, cells deeper within the tissue slice.

There are aspects of Pannexin1 expression within this image region that are only apparent after enhancement in Figure 2(c) (orange) including medium sized round areas with more dense labeling that may be cell bodies whose cell type could be identified with future immunolabeling experiments. Similarly, “squiggles” of more dense

Eigenvalue Spectrum

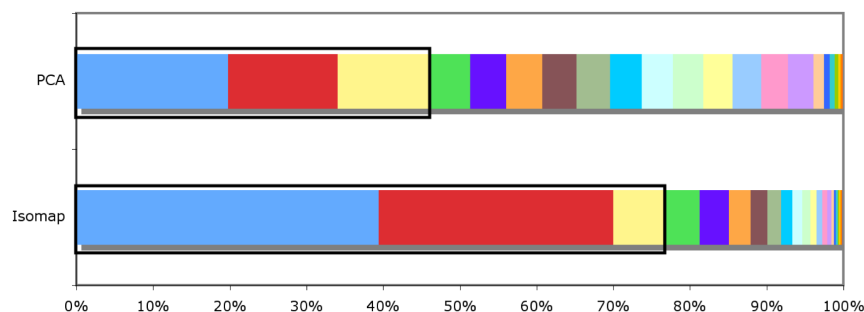


Figure 3. Eigenvalue spectrum from Isomap and PCA. Each color corresponds to one of the eigenvalues, and its size represents its relative magnitude. Larger percentage of first three largest eigenvalues (three from the left) indicates that the three corresponding eigenvectors can better explain (approximate) the original data. The three eigenvalues of PCA sums up to 46.0%; Isomap reaches up to 75.4%.

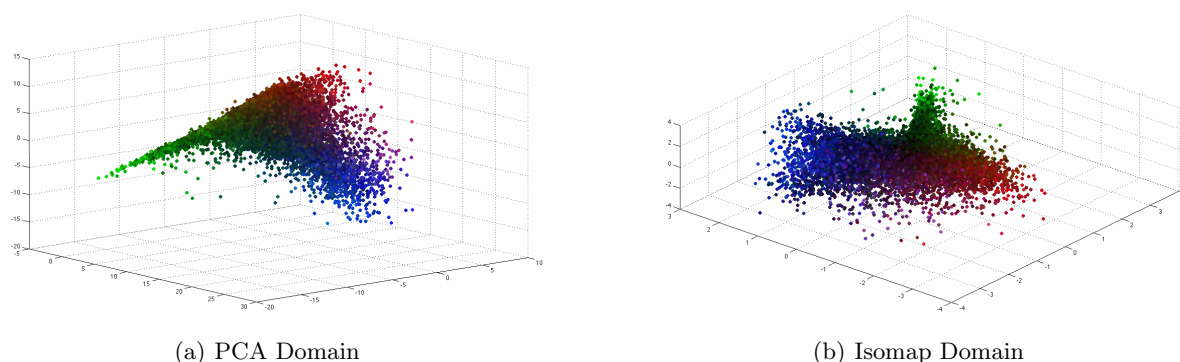


Figure 4. Two domains after reducing the dimensionality of feature vectors into 3D: Figure 4(a) shows the domain generated by PCA; Figure 4(b) shows the domain generated by Isomap.

Pannexin1 expression can be identified on the right side of Figure 2(c) only that they are likely cellular extension coming from these cell bodies and penetrating deeper into the hippocampus.

Figure 2(d), is an enhancement of the Astrocyte cell labeling alone (blue) that highlights their structure. Not only do they cover the outer surface of the tissue, but in Figure 2(d) the contour of these glial cells shows how they also send branching projections deep into the hippocampus in order to establish contact with numerous other cell types known to include neurons. These branching interior projections are extremely difficult to make out in the original image shown in Figure 2(a) and cannot be demonstrated in this way using traditional techniques of adjustments of RGB levels.

The development of novel image rendering techniques like these is crucial to help neuroscientists, like those at NCMIR, show desired aspects of tissue complexity, and new tools for manipulating 3-color images will aid in their ability to dissect the enormous tissue complexity represented by the mammalian brain.

Improving the process of modeling this 3-color protein-labeling data so that it can be meaningfully displayed, annotated and queried by other researchers with the techniques discussed herein will help move the CCDB from the critical function of a data management tool to become a scientific inquiry tool for investigating the structural variation and molecular complexity within the nervous system.

6. DISCUSSION

6.1 PCA vs. Isomap

A common way of measuring the quality of a low dimensional representation is to look at the spectrum of eigenvalues.¹⁷ As we project high-dimensional data into 3D space, the entire data is approximated with only three eigenvectors. A large eigenvalue means that the corresponding eigenvector can better explain the original data, and 100% means it can reconstruct the original data without any error. As shown in Figure 3, the sum of three eigenvalues from PCA achieved 46.0% of that of total eigenvalues. The spectrum of Isomap shows that the top three eigenvalues dominate the entire spectrum, 75.4%.

Figure 4 shows the two domains generated from PCA and Isomap. Both PCA and Isomap were able to cluster voxels at three corners, each of which indicates high intensity values of one channel. From a transfer function designer’s perspective, it was easier to locate features in the Isomap domain than PCA one. The domain from PCA lays out points too closely to each other, resulting distinct areas of interest to display identical colors when a Gaussian widget selects a region in the PCA domain.

6.2 Further Applications

The result of our dimensionality reduction algorithms can be used as a starting point for trial-and-error⁴ approaches. The same set of widgets developed for dual domain interaction can also be used to explore the result. This means that we do not generate one final transfer function. We give a compact form of the domain so that visualization scientists can start with the best domain that can effectively isolate areas of interest when high dimensionality is necessary for the isolation. Many algorithms in transfer function design are geared towards better recognition of features in the data. Gradient information is useful because it can capture the boundary of objects. The domain we get after applying Isomap contains enough data to successfully isolate features - otherwise, we can simply add more features that effectively isolate interesting parts in the image before applying dimensionality reduction algorithms. Then, the dimensionality reduction algorithms can strip out superfluous components in the feature vectors. Thus, the domain represents the best possible low representation in 3D, especially compared to those with only three types of information.

In addition, if advanced interfaces for classification or automatic transfer function generation algorithms^{26–28} are changed to accommodate the domain created here, they might produce better results than using channel intensities only. Furthermore, the low-dimensional representation (3D) is much smaller than the original dimensionality (24D in our example), and thus the space and time complexity for handling this smaller data set can also be kept lower. One advantage over other algorithms is that dimensionality reduction algorithms are unsupervised learning algorithms,²¹ that is, they do not require user input during the computational process, nor prior knowledge about the data from the users, such as pre-classification of the input data or selection in stochastic processes.

6.3 Scalability

Due to recent improvements of confocal scanning technology, the size of the image data that scientists usually handle has reached Giga-pixels and more. The computational cost of most steps in the dimensionality reduction algorithms, such as computing pairwise distance and eigenvectors, or running the shortest distance algorithm on billions of elements are, however, sometimes prohibitively expensive. Although there have been many algorithmic attempts to avoid this barrier,¹⁹ we were unable to process very high resolution images in just a single run.

We cut out a small cube (128 x 128 x 24 voxels) from the entire data volume, containing the most interesting parts for scientists. With the small subset of data, a transfer function can be calculated much more easily. With the transfer function, we extend the result to other parts of data with the following algorithm:

- We compute the feature vector for X , a point in the large volume, and we find 3 nearest neighbors of X , in terms of Euclidean distance in the feature vector space, from the reference data, $\{X_i\}_{i=1}^N$, that we used for transfer function design.

- We estimate the low-dimensional embedding of X by interpolating the three neighbors. We use the same assumption that the Locally Linear Embedding algorithm¹² makes; if X and its neighbors are close enough, then we can assume that X is on the linear plane that its neighborhood points generate. We use the weights to get Y of X by linearly interpolating from $\{Y_i\}_{i=1}^N$.

7. CONCLUSION AND FUTURE WORK

We proposed a new method to build transfer functions for multi-channel data sets. Feature vectors including multiple mathematical properties, as well as each of the channel intensities, were constructed. The high dimensionality of the feature vector does not allow a user to visualize the transfer function domain directly. A recent development in nonlinear dimensionality reduction, Isomap, and a classical linear algorithm, PCA, were used to reduce the high-dimensionality to a manageable size. The algorithms guarantee that the low-dimensional domain retains details of the high-dimensional data set. The major benefit of looking at combined features is that co-occurrence, either a positive or negative correlation, of multiple components in the feature vector can have a different color by isolating the regions from other parts.

This work started in collaboration with NCMIR, but we believe our approach is general enough to be applicable to other data domains.⁴ Adopting our approach to data sets from different disciplines will help us better understand the generality of our approach.

There have been many algorithms proposed for nonlinear dimensionality reduction since two seminal works, i.e., Isomap and LLE, published in 2000. Although we chose the Isomap algorithm for our analysis, we believe that other alternatives may produce similar results. Especially, Locally Linear Embedding¹² (LLE) and Maximum Variance Unfolding¹⁰ (MVU) are immediate candidates for experimentation. Many aspects of LLE are opposite to that of Isomap: sparse matrix computation instead of dense matrix computation, and utilizing locally linear patches instead of geodesic distances. One technical hurdle of MVU can be, however, its computational cost. MVU requires a semidefinite programming solver and it may not scale well with typical sizes of volume data sets. In addition, we are investigating other variants^{13,14} whether or not one of the algorithms can produce better results.

Finally, the computational aspect of the algorithms used is a big challenge. We expect that further optimization of the algorithms and their implementation will alleviate this problem. General purpose GPUs can be used to accelerate parallelizable parts. Subsampled data to find low-dimensional representations, along with a reasonable reconstruction algorithm, should improve overall computational cost.

ACKNOWLEDGEMENT

This publication was made possible by Grant Number (NCRR P41-RR004050) from the National Center for Research Resources (NCRR), a part of the National Institutes of Health (NIH). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the NIH. This publication is based in part on work supported by Award No. US 2008-107, made by King Abdullah University of Science and Technology (KAUST), by NIH Award (NIGMS F32GM092457) and by National Science Foundation Awards (NSF MCB-0543934 and OCE-0835839). Finally, the authors would like to thank Lawrence Saul and the anonymous reviewers for their helpful comments.

REFERENCES

- [1] Puig, A., Tost, D., and Ferre, M., "Design of a multimodal rendering system," in [*Proc. 7 th International Fall Workshop Vision, Modeling and Visualization 2002, Greiner G., Niemann H., Ertl T., Girod B. and Seidel HP. Editors, pp 488-496*], (2002).
- [2] Caban, J. and Rheingans, P., "Texture-based transfer functions for direct volume rendering," *Visualization and Computer Graphics, IEEE Transactions on* **14**(6), 1364–1371 (2008).
- [3] Kindlmann, G., "Transfer functions in direct volume rendering: Design, interface, interaction," *Siggraph Course Notes* (2002).

- [4] Pfister, H., Lorensen, B., Schroeder, W., Bajaj, C., and Kindlmann, G., “The transfer function bake-off,” in [*VIS '00: Proceedings of the conference on Visualization '00*], 523–526, IEEE Computer Society Press, Los Alamitos, CA, USA (2000).
- [5] Kniss, J., Kindlmann, G., and Hansen, C., “Multidimensional transfer functions for interactive volume rendering,” *IEEE Transactions on Visualization and Computer Graphics* **8**(3), 270–285 (2002).
- [6] Levoy, M., “Display of surfaces from volume data,” *IEEE Computer Graphics and Applications* **8**(3), 29–37 (1988).
- [7] Kindlmann, G., Whitaker, R., Tasdizen, T., and Moller, T., “Curvature-based transfer functions for direct volume rendering: methods and applications,” *Visualization* (Jan 2003).
- [8] Hladuvka, J., Konig, A., and Groller, E., “Curvature-based transfer functions for direct volume rendering,” *Spring Conference on Computer Graphics* (Jan 2000).
- [9] Correa, C. and Ma, K.-L., “Size-based transfer functions: A new volume exploration technique,” *IEEE Transactions on Visualization and Computer Graphics* **14**(6), 1380–1387 (2008).
- [10] Weinberger, K. Q. and Saul, L. K., “An introduction to nonlinear dimensionality reduction by maximum variance unfolding,” in [*AAAI*], AAAI Press (2006).
- [11] Weinberger, K. Q., Sha, F., Zhu, Q., and Saul, L. K., “Graph laplacian regularization for large-scale semidefinite programming,” in [*NIPS*], (2007).
- [12] Roweis, S. T. and Saul, L. K., “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science* **290**(5500), 2323–2326 (2000).
- [13] Belkin, M. and Niyogi, P., “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.* **15**(6), 1373–1396 (2003).
- [14] Donoho, D. L. and Grimes, C., “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data,” *PNAS* **100**, 5591–5596 (May 2003).
- [15] Kohonen, T., Schroeder, M. R., and Huang, T. S., eds., [*Self-Organizing Maps*], Springer-Verlag New York, Inc., Secaucus, NJ, USA (2001).
- [16] MacQueen, J., “Some methods for classification and analysis of multivariate observations,” in [*Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*], 281–297 (1967).
- [17] Lewis, J. M., Hull, P. M., Weinberger, K. Q., and Saul, L. K., “Mapping uncharted waters: Exploratory analysis, visualization, and clustering of oceanographic data,” *Machine Learning and Applications, Fourth International Conference on* **0**, 388–395 (2008).
- [18] Tenenbaum, J. B., de Silva, V., and Langford, J. C., “A global geometric framework for nonlinear dimensionality reduction,” *Science* **290**(5500), 2319–2313 (2000).
- [19] Silva, V. D. and Tenenbaum, J. B., “Global versus local methods in nonlinear dimensionality reduction,” in [*Advances in Neural Information Processing Systems 15*], 705–712, MIT Press (2003).
- [20] Borg, I. and Groenen, P. J., [*Modern multidimensional scaling : theory and applications*], Springer series in statistics, Springer (1997).
- [21] Duda, R. O., Hart, P. E., and Stork, D. G., [*Pattern Classification (2nd Edition)*], Wiley-Interscience (November 2000).
- [22] Dagum, L. and Menon, R., “OpenMP: an industry standard API for shared-memory programming,” *Computational Science and Engineering, IEEE* **5**, 46–55 (Jan-Mar 1998).
- [23] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., [*Introduction to Algorithms*], MIT Press and McGraw-Hill (2001).
- [24] Kniss, J., Kindlmann, G., and Hansen, C., “Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets,” in [*VIS '01: Proceedings of the conference on Visualization '01*], 255–262, IEEE Computer Society, Washington, DC, USA (2001).
- [25] DeskVOX: VOlume eXplorer for the Desktop, “<http://www.calit2.net/~jschulze/projects/vox>.”
- [26] He, T., Hong, L., Kaufman, A., and Pfister, H., “Generation of transfer functions with stochastic search techniques,” in [*VIS '96: Proceedings of the 7th conference on Visualization '96*], 227–ff., IEEE Computer Society Press, Los Alamitos, CA, USA (1996).

- [27] Tzeng, F.-Y., Lum, E. B., and Ma, K.-L., “A novel interface for higher-dimensional classification of volume data,” in [*VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*], 66, IEEE Computer Society, Washington, DC, USA (2003).
- [28] Pinto, F. and Freitas, C., “Volume visualization and exploration through flexible transfer function design,” *Computers & Graphics* **32**, 420–429 (Aug 2008).