Original Article

Dimensionality reduction on multi-dimensional transfer functions for multi-channel volume data sets

Han Suk Kim[a, b, *]
Jürgen P. Schulze[c]
Angela C. Cone[b]
Gina E. Sosinsky[b] and
Maryann E. Martone[b]

[a]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.
E-mail: hskim@cs.ucsd.edu
[b]National Center for Microscopy and Imaging Research, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.
E-mails: acone@ncmir.ucsd.edu; gsosinsky@ucsd.edu; mmartone@ucsd.edu
[c]California Institute for Telecommunications and Information Technology, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.
E-mail: jschulze@ucsd.edu

*Corresponding author.

Received: 26 January 2010
Revised: 13 June 2010
Accepted: 17 June 2010

Abstract The design of transfer functions for volume rendering is a non-trivial task. This is particularly true for multi-channel data sets, where multiple data values exist for each voxel, which require multi-dimensional transfer functions. In this article, we propose a new method for multi-dimensional transfer function design. Our new method provides a framework to combine multiple computational approaches and pushes the boundary of gradient-based multi-dimensional transfer functions to multiple channels, while keeping the dimensionality of transfer functions at a manageable level, that is, 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. Applying recently developed nonlinear dimensionality reduction algorithms reduce the high-dimensional data of the domain. In this article, we use Isomap and Locally Linear Embedding as well as a traditional algorithm, Principle Component Analysis. Our results show that these dimensionality reduction algorithms significantly improve the transfer function design process without compromising visualization accuracy. We demonstrate the effectiveness of our new dimensionality reduction algorithms with two volumetric confocal microscopy data sets.

Information Visualization (2010) **9**, 167 – 180. doi:10.1057/ivs.2010.6

Keywords: volume rendering; transfer function; multi-channel volume; light microscopy imaging; dimensionality reduction

Introduction

In volume rendering, the transfer function maps visual properties, such as color and opacity (alpha), to each voxel in a volumetric data set (that is, data filling a 3D space). A carefully designed transfer function can reveal much more visual information than a simple one, and thus is very valuable for scientists exploring 3D volume data sets.[1] Effective transfer functions will identify the different materials, or features of a data set, such as bone or skin in medical data sets, or cell nuclei or protein structures in microscopical images.[2] It is desirable that the transfer function design process be done in real time, to minimize the duration of the partially manual design process.

We call a volume data set *multi-channel* if there are multiple values for each point (voxel) in the data set. The data sets that we consider in this publication are from confocal microscopy images of the mouse hippocampus region, which is a multi-channel data set because it contains the intensities of three different fluorescent antibodies. Our goal is to provide a method to design a transfer function for this kind of multi-channel data set while preserving the information that a multi-dimensional transfer function domain has.

© 2010 Macmillan Publishers Ltd. 1473-8716 **Information Visualization** Vol. 9, 3, 167 – 180
www.palgrave-journals.com/ivs/

The obvious barrier to the goal is the dimensionality of the domain. As an example, suppose that we only use measured intensity and derived gradient magnitude. As we have three measured channels, the dimension of the domain is now six (each measured channel is used as is, plus its gradient magnitude), and the 6D domain is impossible to visualize directly as we do with 2D or 3D transfer functions. One solution can be to consider each channel separately, generating 2D transfer functions for each channel. However, the transfer functions for the three channels would be independent and would not use the information from the other two channels, and thus ignore potentially useful information. Therefore, we propose a method that provides a lower dimensional representation of the original high-dimensional domain, so that users can design multi-dimensional transfer functions while keeping all the rich information in the high-dimensional space.

A previous article[3] proposed a framework that can potentially provide a 3D view of a high-dimensional data space so that we can continue the idea of multi-dimensional transfer function even with multi-channel data. We employed two dimensionality reduction algorithms: Principle Component Analysis (PCA) and Isomap. This article extends the concept of having a low-dimensional representation of the original data domain in two directions. First, we apply another nonlinear dimensionality reduction algorithm, Locally Linear Embedding (LLE), to our problem. There have been many efforts in the machine learning community to propose better algorithms for dimensionality reduction. We decided to use the LLE algorithm because LLE and Isomap are two representative algorithms, but the technical approach is opposite to each other as we will explain in Section 'Dimensionality reduction'. By using two different 'advanced' algorithms, we present the effectiveness of the two algorithms in a field they have not previously been applied to. Second, we investigate our approach with a new microscopy example with different feature vectors. This is to consider two possible scenarios that we often encounter in analyzing volume data sets with transfer function.

Multiple Features: When we have little information about the data and when we do not have enough insight about which features can effectively isolate areas of interest, without our approach one needs to iteratively compare with two or three features. (Then $O(n^2)$ many comparisons are needed where n is the total number of features.) This takes considerably longer, and it is hard to capture the shape or distribution of the entire multi-dimensional domain. As our approach keeps the shape of the multi-dimensional domain while it provides a 3D view of the domain, visualization scientists can easily understand how the voxels are distributed across the multi-dimensional domain. For this scenario, a data set is prepared in such a way that we use up to 24 features, which leads to a transfer function domain with up to 24. This is the domain we then run our dimensionality reduction algorithms. We find that the result from it helps understand the high-dimensional transfer function domain, which is shown by a more effective coloring of the volume data set.

Multiple Channels: Assuming that we have more knowledge about the data and we know that some data channels are more important for the visualization than others. Although many features can possibly isolate more areas of interest, if we are certain about which features we would like to use, then the remaining features may be considered less important. The benefit of dimensionality reduction still holds here because multi-channel data has 3 or 4 intensities and adding one more feature increases the dimension to 6 or 8. The difference compared to the previous scenario is, however, that the original domain is much smaller, that is, 6D in our example. We demonstrate our results with a light microscopy data set of a blood vessel in a mouse's hippocampus region that was stained for specific proteins.

This article is organized as follows: Section 'Related work' discusses previous approaches on transfer function design and dimensionality reduction algorithms. Then, Section 'Dimensionality reduction' reviews the three dimensionality reduction algorithms we employed for our work. Section 'Transfer function design' presents how we apply dimensionality reduction algorithms to automate transfer function design for multi-channel data sets. Section 'Experiment' outlines the experiments validating our approach and Section 'Result' shows results from dimensionality reductions and volume rendering. Finally, Section 'Further application' discusses how our algorithm fits in many existing algorithms and Section 'Conclusion and future work' concludes this article.

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 from both fields related to our approach.

Transfer function

Transfer functions have been an active area of research in the field of direct volume rendering,[1,2] owing to the significance and the difficulty of finding a good transfer function in direct volume rendering. Especially, significant effort was made to extend the domain of transfer functions[4] from 1D intensity-based transfer functions to multi-dimensional ones. The most popular choice for the second dimension has been gradient magnitude.[4,5] 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[6,7] was shown to be useful in non-photorealistic rendering by emphasizing ridges and valleys. Correa *et al*,[8] considered relative sizes of features as an alternative domain. Caban *et al* [9] 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.

Recently, an algorithm using non-parametric clustering was proposed.[10] The clustering of transfer function domain provides starting points for feature identification and volume exploration. Transfer function with clustering will strengthen our approach because dimensionality reduction does not provide information about which part of the domain corresponds to which feature in the volume data.

Dimensionality reduction

Besides the dimensionality reduction algorithms discussed in 'Dimensionality reduction', many alternative algorithms have been proposed. Although PCA has been successfully used for a long time in many fields, recent development in nonlinear dimensionality reduction algorithms have shown great success in analyzing complex nonlinear high-dimensional data.[11–15] 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 original high-dimensional data.

In scientific visualization, self-organizing maps[16] and *K*-means clustering[17] have widely been used for data exploration. Although each algorithm has shown strengths 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 nonlinear dimensionality reduction algorithms, on the other hand, require a minimum number of free parameters, for example, *K* in the *K*-nearest neighborhood algorithm, and they find a low-dimensional representation that preserves relative distances in high-dimensional space. Moreover, nonlinear dimensionality reduction algorithms produce a single global optimal point, as opposed to many local minima in self-organizing maps or in *K*-means clustering.

Lewis *et al*[18] applied *K*-means clustering, PCA, Isomap and Maximum Variance Unfolding[11,12] 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.

Dimensionality Reduction

In this section, we describe the three dimensionality reduction algorithms that we employ, namely, (i) PCA; (ii) isomap; and (iii) LLE. 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 $D > 3$D space and the outputs are the same number of points but in 3D space. Preserved between the inputs and outputs are the distances between data points. Readers are encouraged to see Roweis and Saul,[13] Tenenbaum *et al*,[19] Silva and Tenenbaum,[20] Borg and Groenen,[21] Duda *et al*,[22] for more details.

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

Principle component analysis

The main idea of 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 $\hat{u}$:

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

The solution for this optimization problem is computed by singular value decomposition (SVD).[22] 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 points 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 $\hat{u}_j$ $(j = 1, \ldots, d)$ which achieves maximum variance. The jth component of $Y_i (j = 1, \ldots, d)$ can be easily computed with the inner product of X_i and $\hat{u}_j$, that is, $Y_{ij} = X_i \cdot \hat{u}_j$.

Owing to the simplicity of the algorithm, although 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.

Isomap

Isomap[19,20] is an extension of multi-dimensional scaling (MDS)[21] for finding a low-dimensional representation of samples with 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 nonlinear 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, that is, 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 the 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 OpenMP[23] can be utilized to boost the performance on a multi-core shared memory system. The second step, with Dijkstra's algorithm[24] 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.[20] 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.

Locally linear embedding

LLE[13] is another popular nonlinear dimensionality reduction algorithm. Whereas Isomap tries to preserve geodesic distance in the manifold, LLE preserves locally linear structures. Namely, LLE exploits the relationship between a point and its neighborhood, which is represented as a linear patch around the point. The linear patches approximating the data are analogous to triangles approximating a sphere in computer graphics. Moreover, the locally linear structures are invariant to translation, scaling and rotation. This means that the algorithm has the freedom to take each patch from tangled objects in high-dimensional space and maps into a low-dimensional space.

The algorithm starts with the same step as in Isomap, computing the K-nearest neighborhood K. Then, it finds W_{ij} such that W_{ij} minimizes $|X_i - \sum_j W_{ij} X_j|^2$ with the constraint of (i) $W_{ij} = 0$ if X_j is not K-nearest neighborhood of X_i and (ii) $\sum_j W_{ij} = 1$. Intuitively, X_i lies on the plane, the linear patch, constructed by X_j's, and X_i can be reconstructed by interpolating X_j's with weight W_{ij} with a minimum error. The last step is to find low-dimensional coordinates Y_i by minimizing the following embedding cost function:

$$\Phi(Y) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2 \tag{2}$$

where this optimization problem is a quadratic form for Y_i's and W_{ij}'s are the weights computed in the previous step. The solution for this problem is found by solving, again, an eigenvalue problem. For this problem, however, we need the $d+1$ eigenvectors of matrix $(I - W)^T(I - W)$ that have the smallest positive $d+1$ eigenvalues as opposed to the largest eigenvalues as in PCA and Isomap.

The benefit of LLE is that: (i) the algorithm is straightforward and easy to understand; (ii) it does not require expensive processing to construct an input for the eigenvalue problem, as opposed to shortest path algorithm in Isomap; and (iii) the matrix constructed by step 2 is usually very sparse, which consumes less memory, whereas Isomap produces a dense matrix. However, the computational cost of finding the eigenvectors of the smallest eigenvalues is usually much higher than that of the largest eigenvalues.

Transfer Function Design

In this section, we discuss how the dimensionality reduction algorithms discussed in Section 'Dimensionality reduction'. We start with defining a transfer function and discuss the limitations of current multi-dimensional transfer function. Then, to overcome the limitation, we propose our solution utilizing dimensionality reduction algorithms and explain in detail how our approach works.

Volume rendering

Volume rendering renders 3D data. In order to get a final 2D image, typical volume rendering pipelines consist of three steps: data acquisition, volume rendering and data exploration via transfer function manipulation. Data acquisition step generates the 3D data. Microscopic data in our application is acquired through capturing protiens in the cell, each of which is stained with different uorescence. Once intensities of the fluorescence light is measured, multiple measurements are stitched together to obtain one slice image across a large region. A stack of such slices are merged together to get a 3D volume data set. Thus, one data set is a box shape and each point in the data, which we call voxel, has multiple values representing each fluorescence intensity. The second step, volume rendering, loads the 3D data into the volume rendering system. Users can rotate, translate, or zoom to observe the data. Depending on the opacity of the data, which normally defined proportional to the intensity of each voxel, one may or may not see the details inside the volume. Therefore, the final step, data exploration with transfer function, is a crucial step to visually emphasize interesting parts in the data. Transfer function defines the mapping from intensity values to color and opacity values. Thus, a good transfer function should be able to emphasize areas of interest, such as a range of intensity values or an object in the data set.

High-dimensional transfer function

Among the three steps in volume rendering, we focus on the last step, transfer function design. The simplest way to define a transfer function *tf* for volume data sets is to define a function from the intensity domain to RGB colors and opacity, known as a 1D transfer function:

$$tf : x \in \mathbb{R} \rightarrow (R, G, B, \alpha) \in [0, 1]^4 \qquad (3)$$

where x denotes the measurement intensity. A few examples can be (i) a linear function that maps white to the highest intensity and black to the lowest intensity; (ii) a window function that maps a color only to the intensity of a certain range either as a step function or a triangular function; or (iii) the superposition of multiple step and/or triangular functions. By adjusting the width of step or triangular functions one can eliminate the opacity parts that are not interesting. The limitation of 1D transfer functions is that it can only distinguish different data values from one another, regardless of their neighborhood, and that they do not translate to multi-channel data sets, where every voxel consists of multiple data values (for instance, for multiple fluorescent proteins in confocal microscopy). As the only information used is intensity, 1D transfer functions map two points that have the same intensity always to the same color and opacity. An improvement over 1D transfer functions for single channel data sets is to derive additional data channels from the data by applying mathematical functions, for instance gradient magnitude or second derivatives. Gradient magnitude helps identify boundaries between materials.

Multi-dimensional transfer function[4] overcame the limitation of 1D transfer functions and extended the transfer function domain to a multi-dimensional one. That is, a transfer function is defined in 2D or 3D as

$$tf : (x, y) \in \mathbb{R}^2 \rightarrow (R, G, B, \alpha) \in [0, 1]^4 \qquad (4)$$

where y is defined as the gradient magnitude. The benefit of having additional dimensions is that features of interest can be isolated. Two data values that are mapped to the same visual properties with a 1D transfer function may be located in different places in a 2D transfer function domain, and by mapping different colors to them these data points can be visually distinguished in the volume data set. The remaining question is whether the second axis of the transfer function domain is effective enough in isolating the features the user is looking for. Gradient magnitude, defined as the L_2-norm of the gradient vector, has been proved to be effective in visualizing the boundaries of objects.[25] This shows that by using multiple data channels, even if the other channels are derived from the first one, transfer function design can become easier and organize the data points into clusters which represent a certain feature of the data, for example, the boundary of a cell structure. Hence, feature isolation, or segmentation, is an essential part of transfer functions. There have been many similar attempts using different derived mathematical values for the second axis[6–9] to more effectively pull out features of the data set. The essence of these ideas is that the additional information about the data points can separate features of interest more effectively in thetransfer function domain.

Furthermore, the way of defining a transfer function in the 2D domain reveals another assumption that multi-channel transfer function design relies on. All the widgets used for transfer function design[4] define a region that encompasses many points and they are mapped to the same or similar color. This means that two points close to each other map to similar visual properties, and more importantly, two points in the same object in the volume data set are located close to each other in the transfer function domain. If the selected transfer function domain does not effectively cluster points of similar characteristics, defining transfer functions in this domain will be more difficult than if a more suitable derivation function were used.

PCA has been widely used in the visualization community, whereas Isomap and LLE are relatively new algorithms that can reduce the dimensionality of data to a low-dimensional representation in a nonlinear fashion. All three algorithms reduce the dimensionality while preserving relative distances between points in high-dimensional space. Therefore, two points close to each other in the domain after applying dimensionality reduction algorithms are also close to each other in the original domain. However, we claim that nonlinear dimensionality reduction algorithms are more appropriate for our problem, because Isomap and LLE find the low-dimensional representation with a nonlinear mapping and because the points in the transfer function domain are often distributed in a nonlinear way.

The distance-preserving characteristics of dimensionality 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 object. 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. Another advantage of dimensionality reduction is that the number of features is less restricted in our proposed method and users are free to add additional fields to the transfer 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.

Reduced transfer function domain

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 'Light microscopy volume'. 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. Two sets of data are prepared for our experiments. The first data set, which we refer to as *24D Tissue Edge Data*, is as the name indicates, confocal volume image data of the edge of a section of brain tissue. In this case, the contrast between the tissue with three spectrally distinct fluorescent labels specific for three proteins or nucleic acids and often displayed with three primary colors and the background containing no fluorescent labels is very high. In this data set, 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. As 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, that is $(\frac{\partial u_i}{\partial x}, \frac{\partial u_i}{\partial y}, \frac{\partial u_i}{\partial z})$ for $i = 1, 2, 3$. In addition, discrete Laplacian valuesare appended to capture ridges and valleys as in Ref. that is, $\frac{1}{6}(\frac{\partial^2 u_i}{\partial x^2} + \frac{\partial^2 u_i}{\partial y^2} + \frac{\partial^2 u_i}{\partial z^2})$ for $i = 1, 2, 3$. Finally, two values from first order statistics are added: mean and variance of channel intensity. In our example, $3 \times 3 \times 3$ voxel cubes are used to capture textural characteristics.[9]

One may argue that the feature vector is superfluous in the 24D Tissue Edge Data, 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 dimensionality reduction algorithm will also eliminate the small variance of the domain.

On the other hand, the second data set, which we call *6D Blood Vessel Data*, is simple and straightforward. It has only six fields: intensity and gradient magnitude from three channels. This volume data set is centered around a blood vessel in the hippocampus. This data set is to verify that our method can provide the same way as in Kniss *et al*[4] but with relatively smaller dimensionality. That is, once we acquire experience about which field effectively isolates areas of interest, we can carefully choose only the fields needed for the visualization goals. By reducing the number of fields, one can reduce the computational cost and generate better results as we will describe in Section 'PCA versus Isomap versus LLE'.

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 that 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 went through to create multi-dimensional transfer functions in previous work. Dual domain navigation and coloring widgets[4] 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 variances 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.

Scalability

Owing 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,[20] we were unable to process very high resolution images in just a single run.

We cut out a small cube (128 × 128 × 24 voxels for 24D data and 50 × 50 × 24 voxels for 6D data) 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 faster. 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 three 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. Then we can represent X as a weighted sum of the three vectors $X_{(1)}, X_{(2)}, X_{(3)}$ with the optimization problem:

$$\text{Minimize} \quad \left| X - \sum_{i=1}^{3} w_i X_{(i)} \right|^2$$

$$\text{with respect to} \quad \sum_{i=1}^{3} w_i = 1$$

- We estimate the low-dimensional embedding of X by interpolating the three neighbors. We use the same assumption that the LLE algorithm[13] 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 w_1, w_2 and w_3 to get the low dimensional coordinate, Y, of X by linearly interpolating from $Y_{(1)}, Y_{(2)}$ and $Y_{(3)}$.

Another scalability issue is for LLE. LLE requires computing the eigenvectors that have the smallest positive eigenvalues. Although the rest of the algorithm is relatively simple and does not require solving dynamic programming as in Isomap, we failed to compute the eigenvectors of a 400 000 × 400 000 matrix. Therefore, if one were to use high resolution images as reference images for transfer function design, LLE may not be the best choice for the work. However, the computation of a 60 000 × 60 000 matrix took significantly less time thanks to an efficient numerical method.[26]

Experiment

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.

Light microscopy volume

Here we apply our new transfer function design process to a data set of images acquired by confocal microscopy of immunolabeled mouse brain tissue. The tissue is labeled with three 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 data set was generated at National Center for Microscopy and Imaging Research (NCMIR) and deposited into the Cell Centered Database

(CCDB; http://ccdb.ucsd.edu, accession #MPID6674), a resource for imaging data sets from advanced light and electron microscopies.[27]

The original microscopy product is a mosaic of 176 overlapping image stacks that were acquired and assembled into one mosaic 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 multiple resolutions: 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 the pannexin1 protein, a recently discovered protein thought to be an ATP release channel participating in ATP signaling cascades. It is highly expressed in the brain. As the brain is a highly heterogeneous organ and expression levels and patterns can vary within subdomains, even within the same class of cell, these brain mosaics span an expanse of millimeters of tissue with each tile is recorded at close to the highest resolution achievable by light microscope. In these images, Pannexin1 protein expression (green) is examined in comparison with that of gap junction protein, connexin43, (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 and large in size. 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. A big challenge is presenting this data set in a way that retains both its complexity and protein specificity to optimally convey any novelty and significance within its biological contexts.

Experiment design

We seek three results from the experiment: (i) rendered image quality; (ii) the distribution and shape of transfer function domain; and (iii) eigenvalue spectrums of PCA and Isomap algorithms. We evaluate the image quality of various images generated by the transfer function. As there is no quantitative way to compare the quality of two images generated from two different transfer function domains, we show several rendered images that successfully isolate features of interest. 1D transfer function domain or even 2D domain is often not able to isolate features. Only the domain having features spread out nicely in it can provide a way to highlight those features. For the distribution and shape of the transfer function domain aligns with the first result. We will show the domain generated by dimensionality reduction algorithms and see if that the shape is nicely stretched in the 3D space. If points are located close to each other, it is more likely for points from different areas of interest to be overlapped together. If this happens, it is impossible to isolate two areas. As we will discuss in Section ‘PLA versus Isomap versus LLE’, domain from LLE and other algorithms can only be compared to this result. The last result we will present, eigenvalue spectrums, compares the quality of PCA and Isomap. These two algorithms produce a list of eigenvectors and corresponding eigenvalues. We are interested in the three largest eigenvalues and if those three dominate the spectrum, it indicates that we have achieved small variance loss during the reduction.

Result

We present results from the experiment we designed in Section ‘Experiment’. First, we show the transfer function domains obtained from the different dimensionality reduction algorithms, and we present rendered images of 3D microscopic data, showing highlighted features. Finally, we discuss the difference between the three algorithms.

Transfer function domain

Figure 1 shows the domain of a transfer function of the 24D Tissue Edge Data described in Section ‘Light microscopy volume’. Figure 1(a) is the view from the top of the domain, that is, 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 searches. 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 the 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 go down to +z depth from right to left; region 4 is where ridge voxels are distributed; and region 5 denotes other valleys but in opposite direction to region 2.

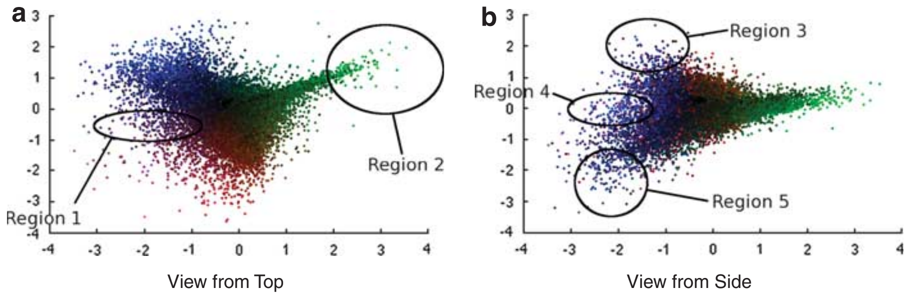

Figure 1: The domain distribution of feature vectors after Isomap dimensionality reduction. (**a**) is the view from the top of the domain, that is, projected to the *xy* plane; (**b**) is the view from a side of the domain, that is, projected to the *xz* plane.

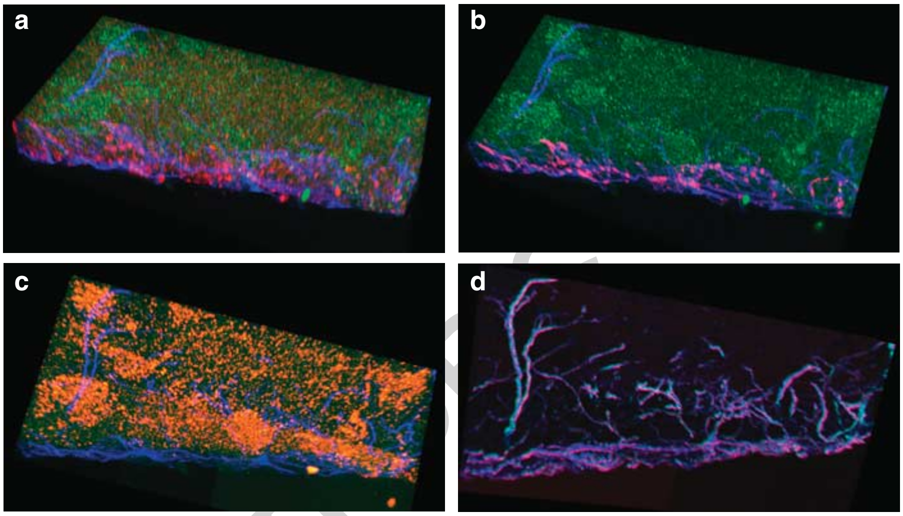

Figure 2: Images of a three-channel 512 × 256 × 24 volume: (**a**) Traditional volume rendering with only three-channel intensity; (**b**) Volume rendering with PCA domain. Connexin43 expression within astrocytes; (**c**) Volume rendering with Isomap domain. Pannexin1 expression; (**d**) volume rendering with Isomap domain. Astrocytes Enhancement. All the images depict the same volume but with different transfer functions. Maximum intensity projection was used for slice compositing. Figures 2(a) is a result with a transfer function that maps the three data channels to red, green and blue, respectively. Figures 2(b)–(d) show examples of our methods. Each of the three images was generated in the domain in which our dimensionality reduction algorithms were applied. Figure 2(b) shows the area where Connexin43 and astroctytes 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.

Many widgets for editing transfer functions, for instance as in Kniss *et al*[25], can be used in the same way with this domain. For the images in Section 'Rendering results', only a Gaussian widget was used. The Gaussian widget in the 3D domain maps point x to a color according to the Gaussian distribution:

$$I(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}|\sigma|^{3/2}} \exp\left\{-\frac{1}{2\sigma^3}(x-\mu)^T(x-\mu)\right\} \quad (5)$$

where μ denotes the location of the center, and points around the center are affected by the widget. σ decides how far from the center the widget's influence reaches. Larger σ values color a wide range of points whereas smaller σ affects only a small range of points. The location of the center, μ, is selected by users, and our tool provides a method to adjust σ. In the future, we may investigate a tool which separately controls σ in three directions, but this widget was effective enough in our experiments.

Rendering results

We rendered two datasets, 24D Tissue Edge (512 × 256 × 24 voxels) and 6D Blood Vessel (300 × 250 × 24 voxels), with DeskVOX,[28] an open-source volume rendering software. A maximum intensity projection algorithm was used for blending the slices. Figures 2 and 3 show several screen shots.

Tissue edge

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 multichannel 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 Figures 2(b)–(d).

The astrocytes (blue in Figure 1(a)) cover and their processes wrap the outer surface of the hippocampus. These astrocytes, as well as other cell types, express Connexin43 (red). Figure 1(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 Pannexin1 expression can be identified on the right side of Figure 2(c) only that they are like 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.

Blood vessel

Figure 3 shows the blood vessel in a hippocampus. Figure 3(a) is from the traditional volume rendering with 1D transfer function, that is, assigning red, green, and blue to each channel, respectively. It is hard to recognize how astrocyte and Connexin43 gap junctions interact each other around the blood vessel mainly because of tissue complexity. Furthermore, because there is no stain inside the blood vessel, it is difficult to recognize where the vessel actually is located.

On the other hand, Figure 3(b) is an enhancement of Figure 3(a) using the proposed method in this article. During the transfer function design, we colored the area where three intensities and three gradient magnitude values are all near zero. That area in the transfer function domain successfully isolated the inside of blood vessel.

Improving the process of modeling this three-color protein-labeling data so that it can be meaningfully displayed, annotated and queried by other researchers with the techniques discussed here will help the CCDB integrate the critical function of data management with data mining tools for investigating the structural variation and molecular complexity within the nervous system.

PCA versus Isomap versus LLE

A common way of measuring the quality of a low-dimensional representation is to look at the spectrum of eigenvalues[18] in PCA and Isomap. Figures 4(a) and (b) show the eigenvalue spectrum of two data sets, 24D Tissue Edge and 6D Blood Vessel data, respectively. Each bar represents the eigenvalue magnitude, and they are sorted from largest to smallest. The first spectrum is the result from 6D space and 6 eigenvalues are generated.

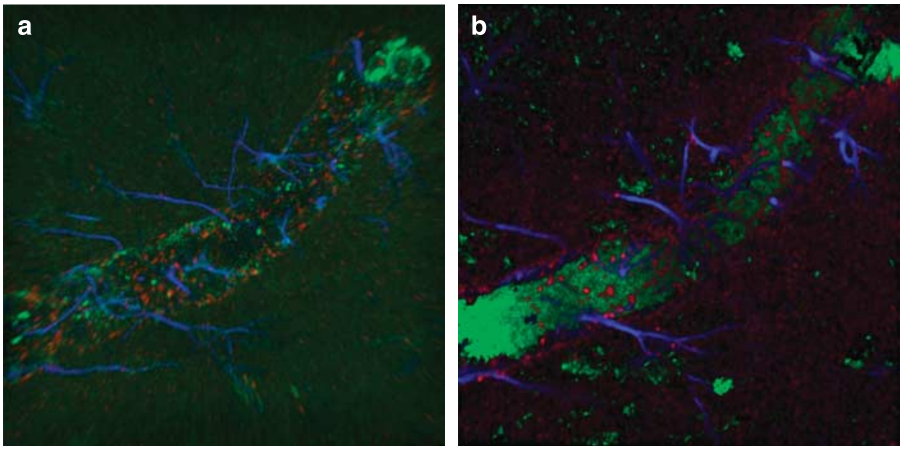

Figure 3: Rendered images of a three-channel 300 × 250 × 24 volume. (**a**) Traditional volume rendering with three-channel intensity only; (**b**) Volume rendering with a transfer function in the LLE domain.

Similarly, the 24D Tissue Edge data has 24 eigenvalues. 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 per cent means it can reconstruct the original data without any error.

Figure 4(b) shows that the sum of three largest eigenvalues produced by Isomap is 94 per cent, which dominates the entire spectrum. Only 6 per cent of the total variance is lost by the dimensionality reduction. In this case, even with the first and second largest eigenvalues, it sums up to 86 per cent and then with this result, one can reduce the dimensionality down to 2D, which is much easier to work with. The spectrum of PCA was not as high as that of Isomap, namely only 80 per cent. However, compared to the 24D Tissue Edge data, it is a promising result. As shown in Figure 4(b), the sum of the three eigenvalues from PCA achieved only 46.0 per cent of that of the total eigenvalues. The spectrum of Isomap shows that the top three eigenvalues dominate the entire spectrum, 75.4 per cent.

We believe that the difference between the 6D Blood Vessel and 24D Tissue Edge data stems from the characteristics of the data. The 24D Tissue Edge data contains more mathematical feature information than 6D Blood Vessel data. Some of them may contribute to identifying features that we are interested in, but the rest may not have played a significant role. Adding more features, or increasing the dimensionality of the domain, makes all three algorithms difficult to reduce to a lower dimension. It is mainly because a new feature spread out the domain without the feature in the new axis. Then the new domain with the new feature has more variance in the distribution than the one without it.

However, we were unable to quantitatively compare LLE with PCA and Isomap. It is because LLE generates the result from a different method, finding the smallest eigenvalues. While PCA and Isomap approximate the high-dimensional vector by cutting off eigenvectors from the observation that eigenvectors with smaller eigenvalues contribute less to the reconstruction of the original vector. However, the eigenvalue spectrum in LLE is not used for this type of approximation, and thus comparing the magnitude of eigenvalues does not reveal any information to us.

There have been many reports on the comparison between algorithms with predefined data[20,29] but none of them was able to compare quantitatively. It is not clear how to measure 'how faithfully' the low-dimensional representation explains the original data. Hence, the only way that we were able to use for the comparison of the result of PCA, Isomap and LLE was a qualitative way, that is, by looking at the domain itself.

Figure 5 lists the two domains generated from 6D Blood Vessel data with PCA and LLE. To help the reader's understanding of the point distribution, we colored them

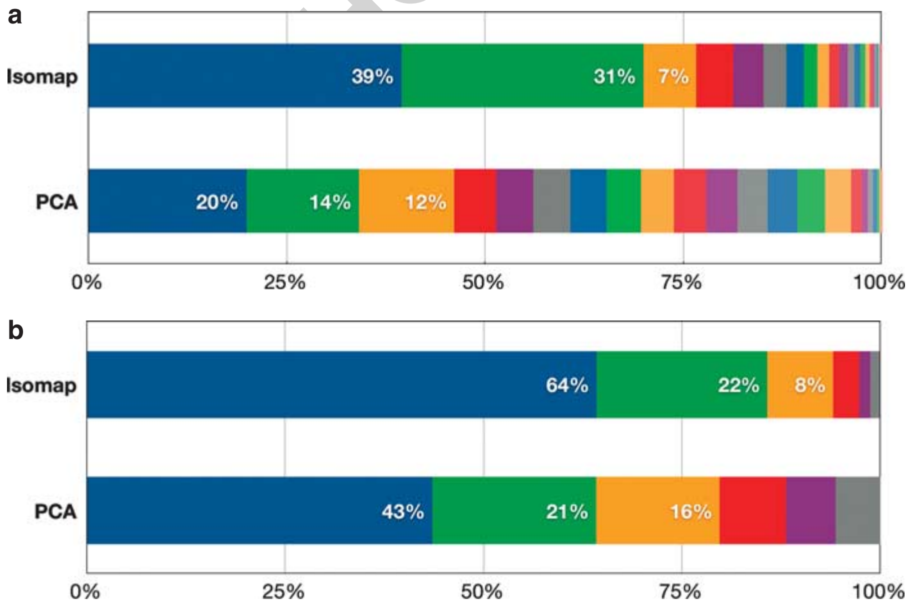

Figure 4: Eigenvalue spectrum from Isomap and PCA: (**a**) Eigenvalue spectrum for 24D Tissue Edge Data; (**b**) Eigenvalue spectrum for 6D Blood Vessel Data. *Notes*: Each color corresponds to one of the eigenvalues, and its size represents its relative magnitude. Larger percentage of first three largest eigenvalues (three from the left) indicates that the three corresponding eigenvectors can better explain (approximate) the original data. With the 24D Tissue Edge data, the three eigenvalues of Isomap sums up to 75.4 per cent and PCA reaches up to 46.0 per cent. With the 6D Blood vessel data, it showed a much better spectrum, 94 per cent with Isomap and 80 per cent with PCA.

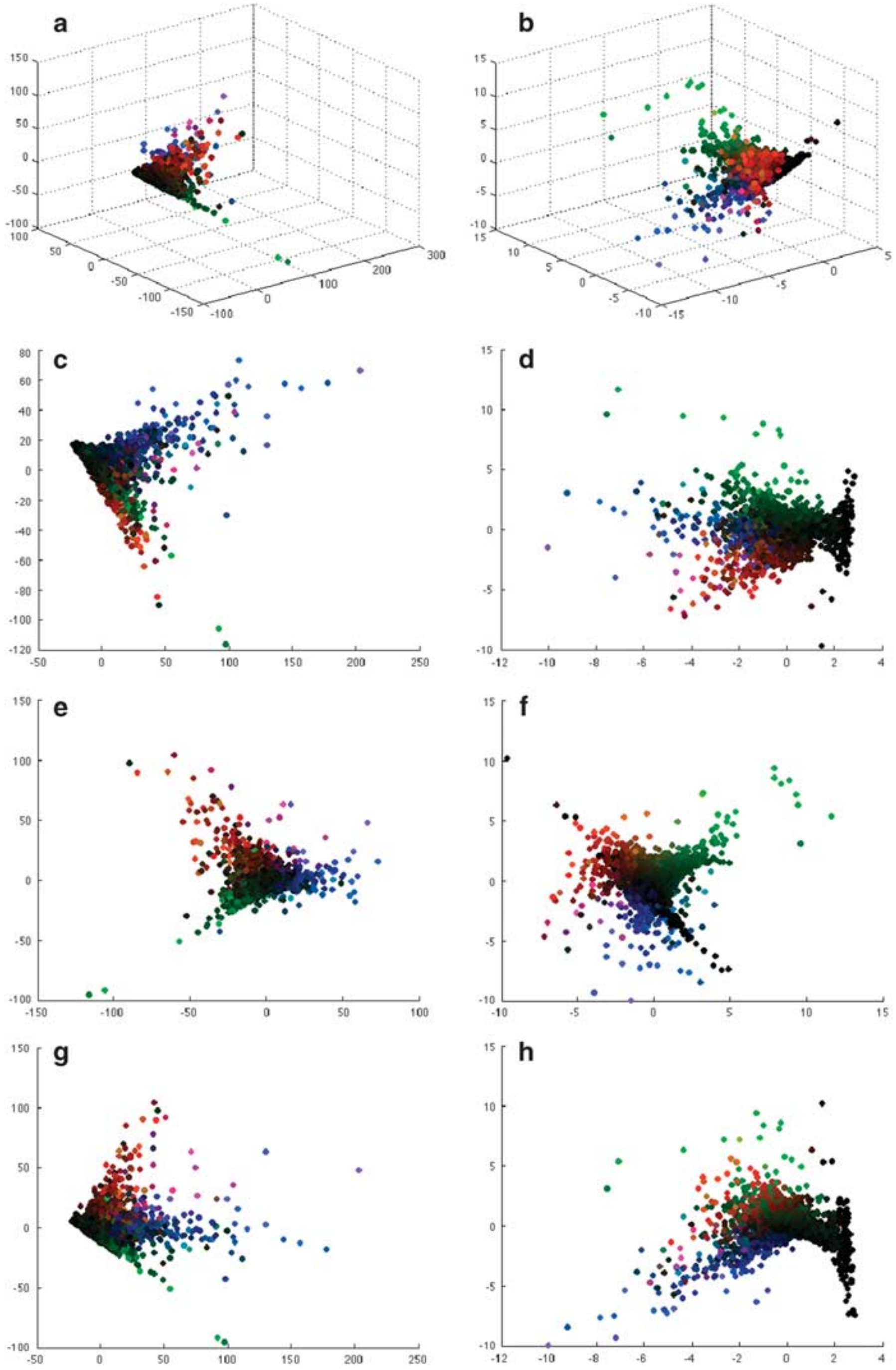

Figure 5: Two domains after reducing the dimensionality of feature vectors into 3D: Figures 5(a), 5(c), 5(e) and 5(g) show the domain generated by PCA; Figure 5(b), 5(d), 5(f) and 5(h) show the domain generated by Isomap. (**a**) PCA domain; (**b**) LLE domain; (**c**) PCA xy projected domain; (**d**) LLE *xy* projected domain; (**e**) PCA *xy* projected domain; (**f**) LLE *xy* projected domain; (**g**) PCA *zx* projected domain; (**h**) LLE *zx* projected domain.

with gradient magnitude of three channels. Furthermore, because it may not clear to understand the shape of 3D distribution, we also list 2D projection of the domains to three major planes, *xy*-, *yz*-, and *zx*-plane.

Both PCA and LLE were able to cluster voxels at three corners, each of which indicates high gradient values of one channel. The shape roughly looks like a tripod. The shape of the PCA domain is simple, many near-zero gradient magnitude values are densely clustered and it 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. On the other hand, the LLE domain spreads out the distribution more clearly, which makes the dual navigation easy.

Further Application

The result of our dimensionality reduction algorithms can be used as a starting point for trial-and-error[1] approaches. One can explore the result with the same set of widgets developed for dual domain interaction. This means that we do not generate one final transfer function. We only give a compact form of the domain. Visualization scientists can start with the compact representation that can still effectively isolate areas of interest. 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 boundaries of objects. The domain we get after applying Isomap contains enough data to successfully isolate features – otherwise, we can simply add more features which effectively isolate interesting parts in the image before applying dimensionality reduction algorithms. Then, the dimensionality reduction algorithm 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[30–32] 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,[22] that is, they do not require user input during the computational process, nor prior knowledge about the data from the users, such as preclassification of the input data or selection in stochastic processes.

Finally, the low-dimensional representation can be further processed for clustering[10] to guide visualization scientists. Clustering the domain gives scientists a rough guideline for where to start. This idea strengthens our approach because the direction of the axes in the reduced domain has no specific meaning. Each axis of the original high-dimensional data represents one data channel. The axes help to understand the correlation or trends between multiple features. Without this information, it may not be easy to understand which part in the 3D domain represents which features. The clustered result will alleviate this inherent limitation of dimensionality reduction as it gives an initial starting point.

Conclusion and Future Work

We proposed a new method to build transfer functions for multi-channel data sets. Feature vectors consist of multiple mathematical properties, as well as each of the channel intensities. The high dimensionality of the feature vector does not allow a user to directly visualize the transfer function domain. We applied two recent developments in nonlinear dimensionality reduction, Isomap and LLE, and a classical linear algorithm, PCA, 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.

We believe our approach is general enough to be applicable to other kinds of data domains beyond the neuroscience imaging examples we show here.[1] Medical imaging community merges multiple data sources such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) and the multi-dimensional transfer function originally came out of MRI data sets. If one wants to consider multi-modal data and to design a transfer function in a multi-dimensional domain, for example intensity and gradient magnitude, the dimension of the total features considered easily exceeds 3D. Our approach can be applied in the same way as we discuss in this article but it would be interesting to study our approach in depth on these types of data sets.

While Isomap and LLE already showed sufficient results for our goal, we are aware that many algorithms have been proposed for nonlinear dimensionality reduction since the two seminal works, Isomap and LLE, published in 2000. It would be worth investigating other variants[14,15] to test whether one of these variant algorithms can produce better results.

Finally, although the dimensionality algorithms run only one time, the amount of computation needed for these algorithms remains a big challenge. However, we expect that further optimization of the algorithms and efficient implementations will alleviate this problem. For example, general purpose graphic processing units (GPU) can be used to accelerate parallelizable parts and the use of subsampled data to find low-dimensional representations, along with a reasonable reconstruction algorithm, should improve overall computational cost.

Acknowledgements

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). The Cell Centered Database is supported by NIH grants from NCRR RR04050, RR RR08605 and the Human Brain Project DA016602 from the National Institute on Drug Abuse, the National Institute of Biomedical Imaging and Bioengineering and the National Institute of Mental Health. Finally, the authors thank Lawrence Saul and the anonymous reviewers for their helpful comments.

References

1 Pfister, H., Lorensen, B., Schroeder, W., Bajaj, C. and Kindlmann, G. (2000) The transfer function bake-off. In: In: T. Ertl, B. Hamann and A. Varshney (eds.) *VIS '00: Proceedings of the Conference on Visualization '00*. Los Alamitos, CA: IEEE Computer Society Press, pp. 523–526.

2 Kindlmann, G. (2002) Transfer functions in direct volume rendering: Design, interface, interaction. In: T.S. Yoo with G. Gerig, R. Whitaker, R. Machiraju and T. Möller (eds.) *Siggraph 2002 Course 50: Image Processing for Volume graphics*.

3 Kim, H.S., Schulze, J.P., Cone, A.C., Sosinsky, G.E. and Martone, M.E. (2010) Multichannel transfer function with dimensionality reduction. *Visualization and Data Analysis 2010* 7530(1): 75300A, SPIE.

4 Kniss, J., Kindlmann, G. and Hansen, C. (2002) Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 8(3): 270–285.

5 Levoy, M. (1988) Display of surfaces from volume data. *IEEE Computer Graphics and Applications* 8(3): 29–37.

6 Kindlmann, G., Whitaker, R., Tasdizen, T. and Moller, T. (2003) Curvature-based transfer functions for direct volume rendering: methods and applications, *Proccedings of the 14th IEEE Visualization 2003 (VIS'03)*, pp. 513–520.

7 Hladuvka, J., Konig, A. and Groller, E. (2000) Curvature-based transfer functions for direct volume rendering. In: B. Falcidieno (ed.) *Spring Conference on Computer Graphics*, pp. 58–65.

8 Correa, C. and Ma, K.-L. (2008) Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics* 14(6): 1380–1387.

9 Caban, J. and Rheingans, P. (2008) Texture-based transfer functions for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 14(6): 1364–1371.

10 Maciejewski, R., Woo, I., Chen, W. and Ebert, D. (2009) Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics* 15(6): 1473–1480.

11 Weinberger, K.Q. and Saul, L.K. (2006) An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *AAAI*. Proceedings of the Twenty-first National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference; 16–20 July, Boston, MA: AAAI Press, http://dblp.uni.trier.de.

12 Weinberger, K.Q., Sha, F., Zhu, Q., Saul, L.K. (2007) Graph Laplacian regularization for large-scale semidefinite programming. In: B. Schoelkopf, J. Platt, and T. Hofmann (eds.) *Advances in Neural Information Processing Systems 19*, Cambridge, MA: MIT Press, pp. 1489–1496.

13 Roweis, S.T. and Saul, L.K. (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500): 2323–2326.

14 Belkin, M. and Niyogi, P. (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6): 1373–1396.

15 Donoho, D.L. and Grimes, C. (2003) Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS* 100: 5591–5596.

16 Kohonen, T., Schroeder, M.R. and Huang, T.S. (eds.) (2000) *Self-Organizing Maps*. New York, New Jersy: Springer-Verlag; Secaucus.

17 MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In: L.M. Le Cam and J. Neyman (eds.) *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. Berkeley, CA: University of California Press, pp. 281–297.

18 Lewis, J.M., Hull, P.M., Weinberger, K.Q. and Saul, L.K. (2008) Mapping unchartered waters: Exploratory analysis, visualization, and clustering of oceanographic data. In: M.A. Wani, X. Chen, D. Casasent, L.A. Kurgan, T. Hu and K. Hafeez. *ICMLA*. Seventh International Conference on Machine Learning and Applications, 11–13 December, Sandiego, CA: IEEE Computer Society, http://dblp.uni.trier.de.

19 Tenenbaum, J.B., de Silva, V. and Langford, J.C. (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500): 2313–2319.

20 Silva, V.D. and Tenenbaum, J.B. (2003) Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15*MIT Press, pp. 705–712.

21 Borg, I. and Groenen, P.J. (1997) Modern Multidimensional Scaling: Theory and Applications. Springer Series in Statistics. New York: Springer.

22 Duda, R.O., Hart, P.E. and Stork, D.G. (2000) *Pattern Classification*. 2nd edn. New York: Wiley-Interscience.

23 Dagum, L. and Menon, R. (1998) OpenMP: An industry standard API for shared-memory programming. *Computational Science and Engineering, IEEE* 5: 46–55.

24 Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001) *Introduction to Algorithms*. Cambridge, MA: MIT Press and McGraw-Hill.

25 Kniss, J., Kindlmann, G. and Hansen, C. (2001) Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In: T. Ertl, K.I. Joy, and A. Varshney (eds.) *VIS '01: Proceedings of the Conference on Visualization '01*. Washington, DC, USA: IEEE Computer Society, pp. 255–262.

26 Friedrich, T. (2002) *Nonlinear dimensionality reduction with locally linear embedding and isomap*. MSc dissertation, Department of Computer Science, The University of Shefeld.

27 Martone, M.E. *et al.* (2002) A cell-centered database for electron tomographic data. *Journal of Structural Biology* 138: 145–155.

28 DeskVOX: VOlume eXplorer for the Desktop, http://www.calit2.net/jschulze/projects/vox.

29 Donoho, D.L. and Grimes, C. (2003) Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proccedings of the National Academy of Sciences of the United States of America-Biological Sciences*. 100: 5591–5596. doi: 10.1073/pnas.1031596100.

30 He, T., Hong, L., Kaufman, A. and Pfister, H. (1996) Generation of transfer functions with stochastic search techniques. In: R. Yagel and G.M. Nielson (eds.) *VIS '96: Proceedings of the 7th Conference on Visualization '96*. Los Alamitos, CA: IEEE Computer Society Press, p. 227ff.

31 Tzeng, F.-Y., Lum, E.B. and Ma, K.-L. (2003) A novel interface for higher-dimensional classification of volume data. In: G. Turk, J.J. van Wijk and R.J. Moorhead II (eds.) *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. Washington, DC, USA: IEEE Computer Society, p. 66.

32 Pinto, F. and Freitas, C. (2008) Volume visualization and exploration through flexible transfer function design. *Computers & Graphics* 32: 420–429.