A Fast Projection Technique and its Applications to Visualization of Large Data Sets

Fernando V. Paulovich[1], Danilo M. Eler[1], Jorge Poco[1], Luis G. Nonato[1],
Charl P. Botha[2] and Rosane Minghim[1]
[1]ICMC/USP, São Carlos/SP, Brazil
{paulovic,eler,jpocom,gnonato,minghim}@icmc.usp.br
[2]TU Delft, Delft, The Netherlands
c.p.botha@tudelft.nl

Abstract

Large multivariate data sets still challenge visualization techniques and their interactive capabilities. In this paper we present a new faster formulation of a high precision projection technique that allows much larger data sets to be handled and interacted with. By using this novel approach, named Piecewise Least Square Projection (P-LSP), we provide new and effective ways of exploring the data set through its feature space. The paper shows the use of P-LSP to help data selection by the user during exploration using coherence amongst data items. Examples are shown for images and for volumes as large as 1,000,000 voxels, with as many as 14 variables per voxel, although the technique can also be used for any data set for which a reasonable feature space can be determined.

1 Introduction

Visualizing large multi-variate data sets presents many challenges. Although visualization and exploration techniques have come a long way reducing complexity by sampling and segmenting data sets in various ways, there are many applications for which evaluating the whole data set is necessary, and, for those, current techniques are limited.

For instance, in the case of volumetric data sets, tuning up the parameters for an adequate 3D visualization and progressive exploration may be time consuming and in many cases frustrating, particularly where various scalar fields or fields of mixed nature (scalar, vector, tensor) are present.

Various researchers have spent effort devising interesting ways to deal with multi-valued volumes interactively. Various of the resulting approaches employ techniques derived from multidimensional visualization. They provide a way of accessing the volume through interacting with its feature space whilst simultaneous visualization takes place in object space.

For most of the available techniques, however, there is need to re-sample in lower dimensions and to select or permute attributes in consecutive or side by side visualizations, causing the need for multiple passes through the whole data set and parameter set before the data can be fully analyzed. This paper proposes a novel way to interact with data in feature space that solves some of the drawbacks of the previous approaches.

In order to do that, we have developed a new formulation of a high precision multidimensional projection technique. The original technique, called LSP, is already capable of mapping data sets of many variables with high precision in a 2D visual space, faster than other known projections. The resulting map is a set of points in 2D in which highly related items are projected in the same neighborhood. Ideally, when the preprocessing is adequate and the original multidimensional space allows, there is a formation of groups of highly related individuals and those are separated from one another. These "properties" of a good projection (grouping and separation) can be reasonably measured. Although very precise and faster than other mapping techniques, LSP is noneffective when dealing with massive data sets due to its high processing time. The new approach presented in this work, called *Piecewise Least Square Projection (or P-LSP)*, brings a significant improvement on the processing time while still keeping the precision of the original LSP method.

Due to its effective processing time, P-LSP turns out a viable alternative for problems involving large data visualization and exploration, as it provides a very intuitive mechanism to manipulating volumes indirectly by mapping voxel's attributes into visual space and then echoing the manipulations to object space. The same process can be applied for images or other spatially sampled data, as is illustrated in this paper. As will be seen, once the feature space is mapped to the visual space (usually the two-dimensional plane), interaction becomes quite natural, enabling to accomplish many different exploratory tasks, such as finding feature patterns of interest.

In the case of very large voxel data sets, we also propose, implement and illustrate a supporting strategy. We first partition the object space through a watershed algorithm, then we project a representative of each watershed basin in feature space to visual space. Interactions with the projections in visual space cause corresponding basins to be rendered back in object space. This results in a very interesting alternative for large data set manipulation.

Therefore, we can summarize the contributions presented in this paper as:

- a fast and high-precision projection technique able to handle large data sets is presented. This new approach is 2 to 3 orders of magnitude faster than its precursor while still preserving the quality of the produced layouts;
- a novel way to interact with data in feature space and assess the results on the object space is provide. This new methodology is particularly interesting to explore and discover patterns on multimodal volumetric data;
- finally, a mechanism based on watershed classification is presented to handle very large volumes, improving interaction substantially.

2 Related Work

Volume visualization is faced with handling both multiple fields (various values of various natures - scalar, vector, tensor) per voxel as well as a large number of data points or voxels. The process of defining proper ways to fully explore a volumetric data set is a major problem for visualization. Selection of regions or objects of interest as well as defining proper transfer functions are part of the tasks the analyst may have to perform in order to extract all the information needed from a volume.

Due to the ability of information visualization techniques to handle multi-dimensional information, there have been efforts to employ them for the analysis of the association between variables sampled in the object space. One of the first examples of explicitly linking information visualization with volume visualization was the WEAVE system [GRW$^+$00]. This idea was further expanded upon by Doleisch et al. who formalized the definition of features that could be used in linked 2-D scatterplots to control a linked multi-variate volume visualization [DGH03]. In both these cases, linked 2-D scatterplots are of two discrete features in the feature space. Blaas et al. enabled interactive 2-D projections of arbitrary linear combinations of features in their work [BBP07]. P-LSP takes this a step further by optimizing a 2-D placement of all dataset samples that takes into account distances in the high-dimensional feature space. In other words, points that are highly related in feature space will be projected to the same neighborhood in 2-D. In this way, *all* features are involved in the projection at the same time, not just a subset. The background of this idea is further explained in the following paragraphs.

Some recent developments in projection based visualizations, which is basically an approach to generate data maps by dimensionality reduction to a visual (2D or 3D) space, have improved the precision of these techniques for abstract data sets with a varied number of attributes. In this work we propose to use an extension of one of those projection techniques as a means to interact with the full set of variables in a complete volume or image, as well as to extend the capabilities of projections techniques for handling larger data sets. We define the concept of a multidimensional projection technique next, and discuss the related work on those.

Consider a data set D containing n data instances represented by feature vectors on a m-dimensional space. A multidimensional projection technique, or point placement technique, can be defined as a function that maps each instance $d_i \in D$ into a graphical element (e.g. point, circle, etc.) embedded on a visual space (1D, 2D, or 3D). On the resulting visual layout, the relative positions of the elements reflect some type of relationship amongst the data instances, such as similarity or neighborhood given by a distance function defined on the m-dimensional space [PM08]. In this way users can employ their visual ability to recognize patterns and structures based on similarity. Exploration is usually driven by the location of groups and sub-groups of elements, that are (ideally) highly correlated if they happen to be ‘projected’ in the same neighborhood.

Two classical examples of projection techniques are *Sammon’s Mapping* [Sam64] and *Classical Scaling* (also referred as *Multidimensional Scaling (MDS)*) [CC00]. The former technique first defines a cost function based on the differences among the distances between the elements in the visual space and the desired distances calculated between the data instances in the m-dimensional space. Then it minimizes this function, employing its gradient in an iterative steepest descent process. In the latter technique, a double-centered distance matrix between all pairs of data instances is defined, and a spectral decomposition is applied to recover the cartesian coordinates of the elements in the visual space. Although these techniques yield highly accurate results in terms of distance preservation, both present high computational complexity, O(n^2), complicating their application on large data sets.

Another well known strategy to create multidimensional projections, originally defined as a graph drawing heuristic, is *Force-Directed Placement (FDP)* [FR91]. In the FDP model, each instance is modeled as a particle and its position is determined as the place where the sum of forces acting over it, generated by all other particles (instances), is zero. These forces are proportional to the difference amongst the desired distances between the visual elements and the current distances. The final layout is obtained changing the position of the particles until the system reaches an equilibrium state. Since each instance is affected by all other instances, one iteration of the FDP model is O(n^2). In order to reduce such complexity, Chalmers [Cha96] defined a linear iteration approach where the forces were determined using samples instead of the whole set of instances. However, as n iterations is needed to produce a final layout, this is still a computationally expensive O(n^2) technique.

Aiming at reducing such complexity, Morrison et. al. [MRC02, MC04] create a *Hybrid Model* approximating the Chalmers technique. In this approach a small sample of the data instances is first projected to the visual space. Then the remaining ones are interpolated considering the most similar sampled instance. It reduces the complexity to O($n\ \sqrt[4]{n}$). This process was further optimized by Jourdan and Melançon [JM04] yielding an O($n \log n$) technique. Although such approaches really reduces the complexity of the original model, the produced layouts normally present low similarity or neighborhood preservation, specially for high dimensional data sets, due to the employed approximations. The technique proposed here is also based on approximations, however we seek to maintain a balance between the quality of the projections and the final computational complexity.

Recently, a new approach, called *Least Square Projection (LSP)* [PNML08], was presented which tends to produce good results in terms of neighborhood preservation in a reasonable amount of time. In LSP, differentiating it from most alternative techniques, the similarity preservation is pursued on small neighborhoods and between groups of highly related instances, this being a suitable for high-dimensional data sets. However, its application to large data sets is limited due to the employed process to calculate the neighborhoods and the final placement of the visual elements. The technique presented here, named *Piecewise Least Square Projection (P-LSP)*, is an extension of LSP in order to enable the handling of larger data sets. In the following sections, the bottlenecks of the LSP are identified. We show how P-LSP addresses these, thus reducing its computational complexity and running times without sacrificing too much the quality of the resulting layouts.

3 From LSP to P-LSP

This section presents the new mapping technique called *Piecewise Least Square Projection (P-LSP)* method, which can be seen as a refinement of the *Least Square Projection (LSP)* method.

3.1 LSP Overview

The Least Square Projection technique relies on the assumption that each element p_i of a data set D can be written as a convex combination of its nearest neighbors in the mapped domain, as Euclidean plane for example. In more mathematical terms, let $N_i = \{p_{i_1}, \ldots, p_{i_{ki}}\}$ be the set of ki nearest neighbors of p_i (we are assuming a distance measure is defined in D), and denote by $(x_{p_{i_j}}, y_{p_{i_j}})$ the coordinates of each element $p_{i_j} \in N_i$ when mapped to $\mathbb{R}^2$. Therefore, the two-dimensional coordinates of p_i can be computed by:

$$(x_{p_i}, y_{p_i}) = \sum_{p_{i_j} \in N_i} \alpha_{ij}(x_{p_{i_j}}, y_{p_{i_j}}) \qquad (1)$$

where $\alpha_{ij} > 0$ and $\sum \alpha_{ij} = 1$.

Each element in D gives rise to a vectorial equation as described in (1), which can be assembled into two homogeneous linear systems:

$$L\mathbf{x} = 0; \quad L\mathbf{y} = 0 \qquad (2)$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors representing the x and y coordinates of the mapped elements and L the matrix derived from equation (1).

The sets N_i define a *Nearest Neighbors Graph (NNG)* of D, that is, a graph connecting each element in D to its nearest neighbors. It can be shown that the rank of L is $n-q$, where n is the number of elements in D and q is the number of connected components making up the NNG [SCO04]. Thereby, in order to ensure a single non-trivial solution for the linear systems (2), the NNG should have only one connected component, which can be ensured by adding new edges linking disconnected components of the NNG.

The lack of geometric information in (2) might lead to solutions that are difficult to interpret and analyse. LSP deals with this problem by adding *control points* with geometrical information (constraints) to the systems (2). Control points are representative data instances carefully chosen to stand for groups of highly correlated elements. The Cartesian coordinates of control points are obtained through high precision placement techniques such as *Classical Scaling* [CC00]. Since the control points comprise just a fraction of the original data, the high computational cost of convencional “projection” techniques is not a major issue.

The set of n_c control points lead to new linear systems $H\mathbf{x} = \mathbf{b_x}$ and $H\mathbf{y} = \mathbf{b_y}$, where H is a rectangular $n_c \times n$ matrix whose rows have an entry equal 1 in the corresponding control point column and 0 in the remaining entries; $\mathbf{b_x}$, $\mathbf{b_y}$ are the vectors containing the cartesian coordinates of the already mapped control points. The new systems are coupled to (2), resulting in rectangular systems $A\mathbf{x} = \mathbf{b}$, where $A = \left(\begin{smallmatrix} L \\ H \end{smallmatrix}\right)$ and $\mathbf{b} = \left(\begin{smallmatrix} \mathbf{0} \\ \mathbf{b_x} \end{smallmatrix}\right)$ (the same is valid for y coordinate).

The new full rank rectangular systems are solved in the least-square sense, that is, one finds $\mathbf{x}$ that minimizes $||A\mathbf{x} - \mathbf{b}||^2$. In practice, such minimal solution can be obtained by solving the normal equations

$A^T A\mathbf{x} = A^T\mathbf{b}$ (the same for y) by iterative solvers such as conjugate gradient [She94].

The LSP framework described above has three main drawbacks: (1) the NNG construction may generate a graph with multiple components, resulting in rank-deficient normal equations. Therefore, new edges linking the disconnected components must be inserted in order to ensure single solution. However, the new inserted edges might make uncorrelated elements neighbors, negatively impacting the final two-dimensional mapping; (2) nearest neighbors computation is by far the most costly step of LSP, impairing its usage in application involving large data sets; (3) for massive data sets, the systems solution can be memory and time consuming, impairing to use LSP in large scale problems or time varying data sets.

Aiming at overcoming the aforementioned problems, a new framework called *Piecewise Least Square Projection (P-LSP)* is presented, which is fully described on the next sections.

3.2 Piecewise Least Square Projection (P-LSP)

The reasoning behind P-LSP is to partition the original data set in smaller subsets in which the LSP framework can be efficiently applied. However, the data partition strategy gives rise to a series of issues that need to be addressed in order to make the piecewise approach as effective as the original LSP.

A first issue to be considered is how to partition the data set so as to keep similar instances grouped together while still ensuring a balance of the number of elements in each subset. Many different techniques have been proposed to partition data sets taking into account similarity and partition size. In the present work we opt to employ the so-called *bisecting k-means* [SKK00], since in dense data sets such iterative technique behaves quite similarly to other mathematically well founded partition methods, such as PDDP (which is based on SVD decomposition) [SB04], generating well balanced grouping comprised of similar elements. Furthermore, bisecting k-means avoids the burden of building and solving matrix decomposition, increasing its effectiveness in applications involving large data sets. In our implementation the subdivision is carried out until $k = \sqrt{n}$ groups are reached, since it is an upper bound for the number of groups in a data set [PB95], and produces a number of elements LSP can process efficiently.

The coherent mapping of subsets is another issue to be addressed. In fact, if the conventional LSP framework is applied to each subset independently, no guarantee can be given to ensure that correlated groups will be mapped close to each other in the final two-dimensional space. However, by handling the control points properly one can attain a global relationship among groups without losing the local processing benefit. More specifically, the global relation can be built as follows: Let C_{P_i} be a set of control points chosen from a subset P_i of the partition. In the LSP framework the points in P_i will be mapped to $\mathbb{R}^2$ taken as basis the geometry supplied by the control points in C_{P_i}. Consider now the set $C_P = C_{P_1}, \ldots, C_{P_k}$ comprised by the union of the control points picked out from all groups making up the partition. The set C_P can be seen as a new data set containing a fraction of the initial data, thus LSP can efficiently be applied to map the elements of C_P to $\mathbb{R}^2$. Therefore, the geometrical information assigned to the control points of a particular partition group has been computed in harmony with the control points of other groups, reintroducing the global correspondence among elements lost during the partition stage. The global control point mapping ensures that each group P_i will be mapped in accordance with the mapping of other groups, keeping similar groups close and putting apart dissimilar elements. In practical terms, $n_i = (|P_i|/n) * n^{0.75}$ control points ($|P_i|$ is the cardinality of P_i) are picked out in each partition group P_i. This expression for the number of control points has been reached through experimentations that took into account the mapping quality and computational efficiency.

The P-LSP as discussed above is also shown in compact form in Algorithm 1.

4 Results

The original formulation of LSP has been used in mapping data sets of various kinds, such as document collections, time series and sets of images with consistent level of success in terms of neighborhood preservation as well as appropriate group formation and group separation. For those cases, P-LSP has shown to be able to handle much larger collections, maintaining similar precision to the original formulation (see precision analysis at the end of this Section).

In this section we present the P-LSP approach as an alternative to handle exploration of multi-variate data sets of large quantity of units. We do so by demonstrating some results in visual analysis through similarity relationships between pixels (for images) and voxels (for volumes).

These examples show the effectiveness of a projection technique to help users attain insight on how different choices of pixels/voxels representations (features) affect the similarity between them, and how that similarity can support locating regions and items of interest inside the object under study. In addition we seek to demonstrate the performance of P-LSP for large data sets, and how the approximation employed affects the neighborhood precision of the final projection.

We start by presenting some results on pixel projection.

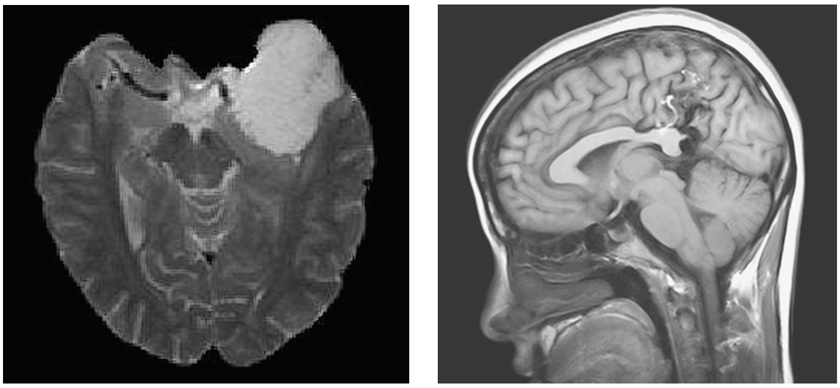
Algorithm 1 Piecewise Least Square Projection (P-LSP).

```
input:   - $D$: data set to be projected.
output:  - the projection of $D$.

procedure $PLSP(D)$
Create the partitions $P = \{P_1, \dots, P_k\}$ using the bisecting k-means algorithm, where $k = \sqrt{n}$.
Set $C_P = \varnothing$

for all $P_i \in P$ do
    Split $P_i$ into $n_i = (|P_i|/n) * n^{0.75}$ clusters, $P_i = \{P_{i_1}, \dots, P_{i_{n_i}}\}$, using the bisecting k-means algorithm.
    Set $C_{P_i} = \varnothing$
    for all $P_{i_j} \subset P_i$ do
        $C_{P_i} = C_{P_i} \cup \{w_{P_{i_j}}\}$, where $w_{P_{i_j}}$ is the medoid of $P_{i_j}$.
    end for
    $C_P = C_P \cup C_{P_i}$.
end for

Map $C_P$ using the original LSP.

for all $P_i \in P$ do
    Map the partition $P_i$ using the original LSP considering the geometry imposed by $C_{P_i}$ .
end for
```

4.1 Pixel projection

Figure 1 shows the images that are used for the examples in this section. Figure 1(a) shows an axial slice of a brain presenting a tumor, and Figure 1(b) shows a sagittal slice of a (different) head.

In an image, an individual data point is given by a pixel. Before applying P-LSP to explore an image, we have to transform each pixel of an image into a set of features, so that a vector space is built for all pixels. This step is followed by a similarity calculation between the pixels. Here we employ two different approaches. In the first, 12 intensity features are extracted. First, we calculate the mean, entropy and standard deviation for the whole image using a $3x3$ mask. Then, for each pixel, we take the mean, variance and standard deviation for each of the three calculations plus intensity from the original image, also using a $3x3$ pixel neighborhood. We add to this set the pixel position (x and y coordinates),



(a) 263x245 pixels.

(b) 992x927 pixels.

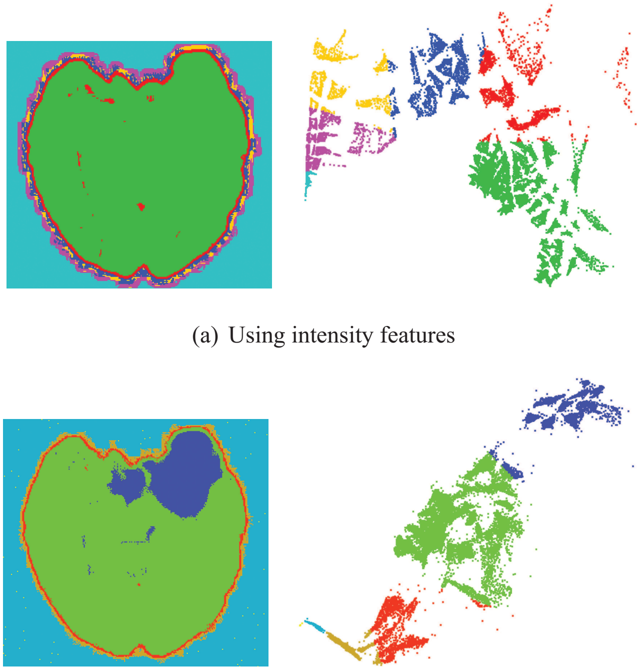
Figure 1: Images for the examples. a) Axial slice of a brain b) Sagittal slice of a head.

resulting in a total of 14 features describing a pixel. In the second approach, we extract 30 Gabor features [JF91, MM96] for each pixel, all within a $9x9$ pixel neighborhood. The 30 Gabor features are defined by varying the frequency $(0,2,4,8,16,32)$ and orientation $(0,\pi/3,\pi/6,\pi/2,3\pi/4)$ of the filter.

In order to compare the effect of these different feature extractions, first we create features for the image in Figure 1(a) and generate different projections for the resulting data sets, that is, we project the pixels on the 2D visual space using an Euclidean similarity measurement over their features. Then, we apply the k-means clustering algorithm [Mac67] to group the projected points in $\mathbb{R}^2$. Finally, points on the projection and corresponding pixels on the image are colored according to the cluster they belong to. The results are presented in Figure 2. In both projections, the background, the brain and the boundary between them are well separated, with result using Gabor features being a little better. However, in terms of detecting the tumor, Gabor features are far better, and the projection based on them separates well the tumor as a set of points, while the projection using the intensity features fail to produce clear separation of the tumor. Interaction is done on the projection with effects shown on top of the image as color or highlight of some sort.

What these pictures show is that, considering a task (e.g. segmentation of classification) which seeks to identify this kind of tumor on images, employing Gabor features may lead to better results when compared to intensity features. Our method can of course be used with other distance functions that measure other types of distances between the employed feature vectors.

Another application where projections of pixels can be applied is for helping users understand the relationships between the results of a pixel-based image segmentation or classification algorithm and the set of features employed. Figure 3 shows an example of that. First we use the k-means algorithm to segment the image presented on Figure 1(b), considering the Gabor features extracted from it. Using the same features we create a projection, coloring its points according to the segmentation. Observe that there is a



(a) Using intensity features

(b) Using Gabor features.

Figure 2: Projections of different sets of features extracted from the same image. Pixels and corresponding points on the projections are colored the same, based on automatically extracted clusters. It is fast to note that Gabor features separates better the tumor from the brain when compared to intensity features.

match between the groups of points in the projection and the elements defined by the segmentation. That is, there is a clear separation between the colors of each segmented part in the final projection. This is an expected outcome since the k-means segmentation splits the image grouping the most similar pixels, and a projection technique defines visual representations which aims at close placing the most similar elements and far apart the dissimilar ones. In cases such as these, our approach can be used to converge to proper feature extractions and similarity calculations for support to other automatic or interactive tasks.

The similarity of various structures according to the employed feature space can be easily seen in the P-LSP projection. The further points are spatially apart, the less similar they are. The P-LSP projection in fact implicitly optimizes for this. For example, the blue skull in the image is very dissimilar from the green background: In the P-LSP projection, these two colors are separated by large distance.

Just as is the case for images, volumes can be analyzed through projections in terms of their most basic components, the voxels. In the next section we present some results in this regard.

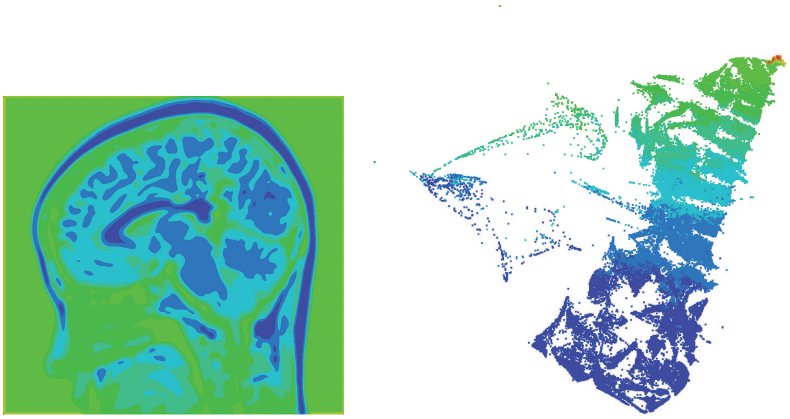


Figure 3: P-LSP projection of the Head image. Colors are based on the segmentation of the image into 10 parts using the k-means algorithm. The projection indicates the similarity relationships between different segments.

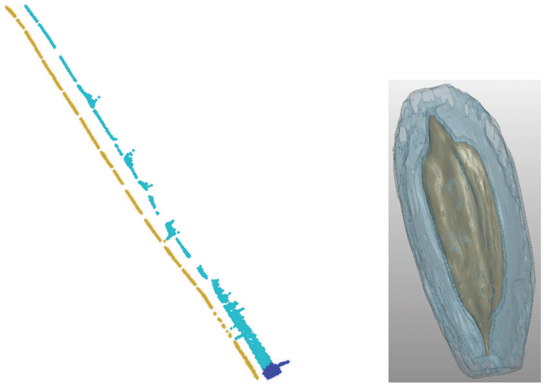
4.2 Voxel projection

As for pixels in images, the first step in applying P-LSP for the visual exploration of voxels on volumes is to compose their feature vectors. For data sets with more than one sampled scalar field (or even vector fields for which magnitude bears significance), the values themselves could be used as features. In our first example, the set of features include a voxel position in the volume (x, y and z coordinates), its intensity and its gradient in a $3x3x3$ neighborhood, resulting in 5 different features.

Figure 4 presents an example of a P-LSP projection generated using this type of features extracted from a Papaya volume. A threshold filter was applied to help distinguish the colors of the core and the pulp. In Figure 4(a), three different groups of points, easily noticed on the projection, were manually selected. Figure 4(b) presents the surface fitting of this volume with the voxels colored according to the groups of selected points. The light blue points represent the papaya outer shell, and the beige ones are its inner volume. The dark blue points represent the background voxels – the colors in the volume are not exactly the same as in the projection due to the transparency employed. There is a very clear separation between the outer shell and the inner volume in the projection, indicating that the type of features employed to represent a voxel is suitable to split these parts. We have used the Euclidean distance between vectors to define the dissimilarity amongst voxels. The surface fitting in Figure 4(b) was generated using the DeVIDE visualization software [BP08].

A second example of voxel visualization is presented in Figure 5. To create these projections we use a sub-set of the original Head CT volume[1], considering slices 30 to 97. The remaining slices do not con-

[1] The head volume is available at `http://www9.informatik.uni-erlangen.de/External/vollib/`



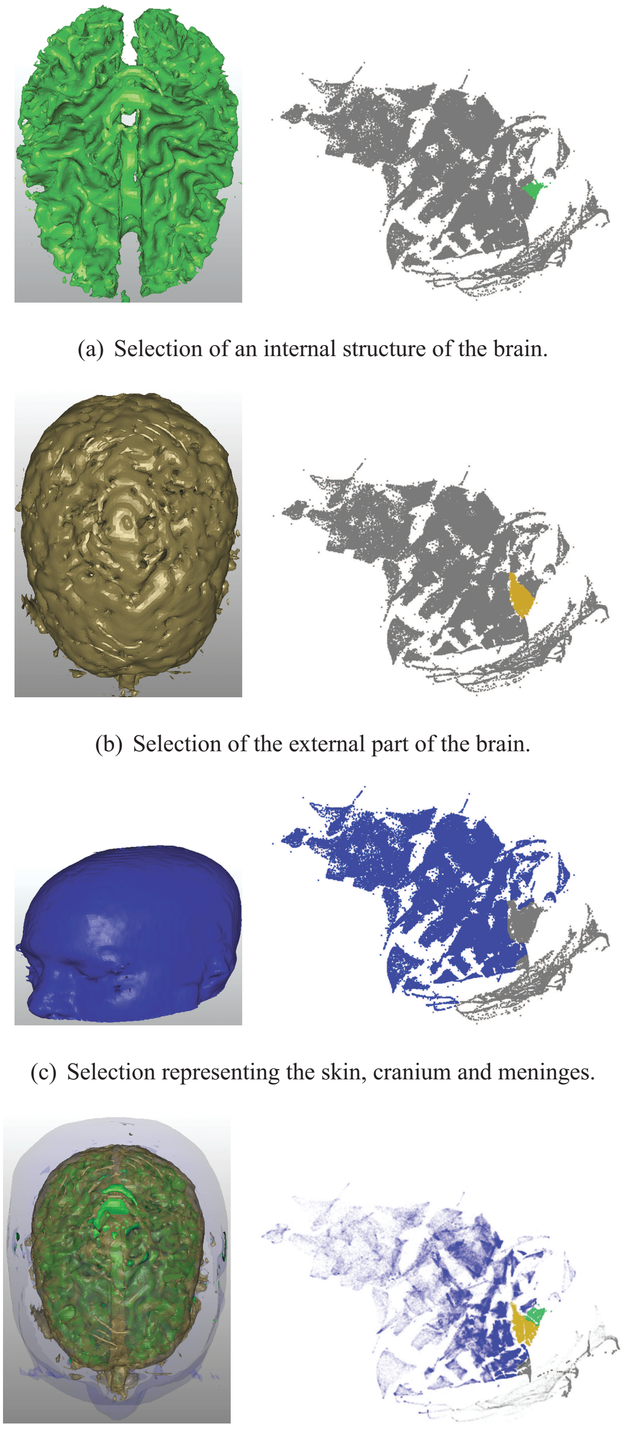
(a) P-LSP projection. (b) Surface fitting.

Figure 4: P-LSP projection from a papaya volume. The projection shows that the employed features separates well the outer shell of the papaya from its inner volume.

tain any voxels with information, they are only background. The resulting sample represents 1,114,112 voxels. Differently from the papaya volume, we use intensity features plus the voxel intensity and its gradient, resulting in 14 features. Figure 5(a) and Figure 5(b) show user selections representing a structure inside the brain and an external part of it, respectively. The selection of points that represent the skin, the cranium, and the meninges can be seen in Figure 5(c). Finally, Figure 5(d) shows all selected parts together. Even for a larger data set, the P-LSP was able to separate well the different structures present in the volume. Additionally, the projections indicate that the set of features employed to represent the voxels deem the internal structure and the external part of the brain very similar since these are closely positioned.

Once there are more points on a large projection than available screen space (we are using a conventional computer monitor), the overlap amongst points is unavoidable, impairing our perception of density in different areas of the projection and of the frontier between them. In order to reduce this problem, we use transparency when drawing the points on the projection presented on Figure 5(d). On this figure, we can verify that the structure inside the brain and its external parts are very dense, and the density of the blue points vary through the projection. It indicates that the employed feature space coupled with the similarity measurement define that the voxels representing the brain parts (in beige and green) are more similar internally than the voxels representing the skin, cranium and meninges (in blue) since they are more dense on the projection. In addition, it is possible to locate groups or sub-groups of high-related voxels on each colored area, which are not visible on the other projections, supporting more refined exploratory tasks.

A different example of applying P-LSP to explore volumetric data sets is shown in Figure 6. In this example we show the result of interacting on the projection of timestep 30 of the Visualization Con-



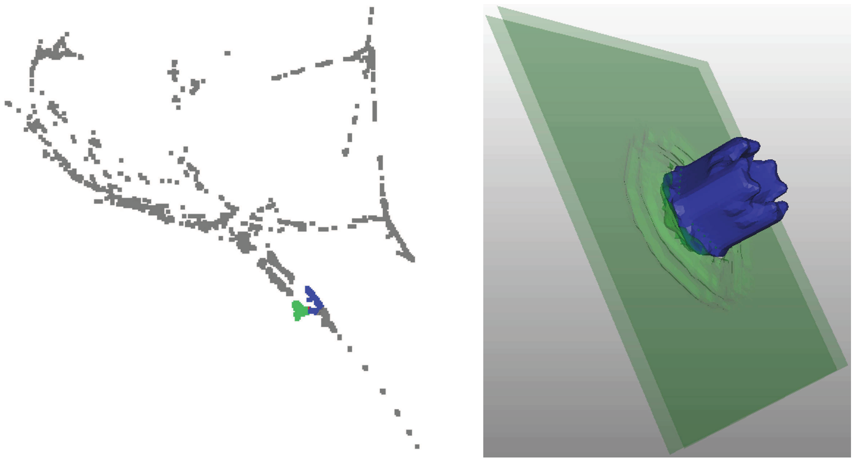
(a) Selection of an internal structure of the brain.

(b) Selection of the external part of the brain.

(c) Selection representing the skin, cranium and meninges.

(d) All above selections together.

Figure 5: P-LSP projections from a sampled head volume. The projections shows a good separation between the different structures of the head.



(a) P-LSP projection.

(b) Surface Fitting.

Figure 6: Visualization of a timestep of the Visualization 2008 data set. Even for a more complex multimodal volumetric data set, P-LSP groups well the inherent structures.

test 2008 data set [WN08]. This data set represents the simulation of the propagation of an ionization front instability and it includes 10 different attributes, the density, temperature, and mass abundances of eight chemical species in more than $36,000,000$ sample points. The huge size impairs conventional exploratory techniques to handle this volume successfully. For performance reasons, we spatially resample this data splitting the volume into $4x4x4$ pieces getting the average of their attributes to represent a new point on the sample, thus reducing the sampling to 576,600 points. Figure 6(a) presents the projection of this data set. In this figure, two different groups of points are selected and colored. Based on this selection we generate the surface fitting (Figure 6(b)) presenting the selected voxels. This shows that even for a more complex multimodal volume data set the P-LSP groups well the inherent structures present on the volume.

Although P-LSP can handle large data sets such as these, and transparency helps to overcome the problems related to point cluttering, as the number of projected points increases further, the response time of interacting with the projections becomes too high for real-time applications. In the next section we present a solution to decrease the number of projected points in feature space, whilst still representing and visualizing all voxels in object space.

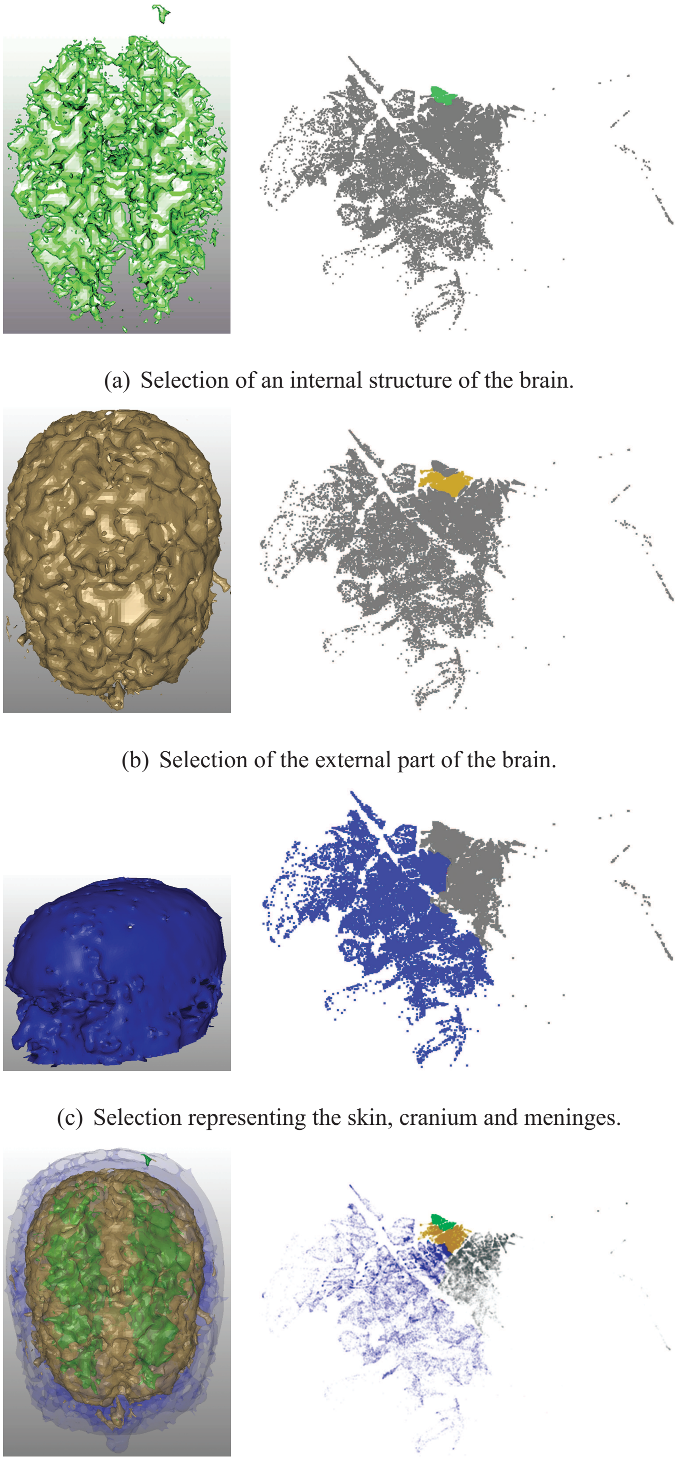
4.2.1 Watershed-based projection

Our strategy to decrease the number of points to project is to first segment the volume using the watershed algorithm [VS91, HO93], and then to project the generated basins (the segmented parts) instead of the voxels themselves. That is to say that there will be one point per basin in the resulting projection, instead of one per voxel. We do not make use of the marker-based watershed, or of any other post-watershed merging, so the basins are small parts of the volume containing pixels that belong to similar structures. To extract image features of a basin, we employ it as a mask on the original volume, extracting intensity features for the set of voxels covered by it. Then, the mean, deviation and variance are extracted from the entropy, mean and deviation of this set of voxels, defining the features of a basin.

Figure 7 presents the projections of basins for the head volume. The number of projected instances is reduced from 1,114,112 voxels to 65,279 basins. Figure 7(a) and Figure 7(b) show the projection of the internal structure and external parts of the brain, respectively. Figure 7(c) presents the selection of the basins that mostly compose the skin, cranium and meninges. Finally, Figure 7(d) shows all selections together. In this last figure we employ transparency to reveal the density of the basins on the projection. The projection of the basins, as well as the projection of the voxels (Figure 5(d)), separate well the brain parts from the skin, skull and meninges, defining them on a more compact and dense area on the projection. It indicates that the features extracted from the basins are similar to the features extract from the voxels in terms of the differentiation between the brain on other parts of the head.

Depending on various filtering parameters, watershed basins sometimes group voxels that could be considered unrelated, creating artifacts in the surface reconstruction considering the selection of basins in the projection. This can be clearly seen comparing Figure 5(c) with Figure 7(c). In this case we have performed a slight amount of basin merging before applying the P-LSP. These artifacts can be reduced by applying even less merging. However, it increases the number of instances (basins) to be projected. Therefore, care must be taken in defining the parameters used for the watershed segmentation so as not to generate too few basins, which can impair the quality of the surface reconstruction, or too many basins, which may increase the interaction time with the projection. Still, although the quality of the view is not as sharp as the one produced by projecting all voxels, using the watershed partitioning together with the projection can definitely decrease computational costs and analysis time previous to the final visualization. This approach also represents a good alternative to complex algorithms for complete basin merging that make decisions automatically on what is the best way to combine different basins, leaving that processing out of the initial phases of analysis.

The times taken to run P-LSP on large volumes are quite motivating. The following Section discusses computational costs.



(a) Selection of an internal structure of the brain.

(b) Selection of the external part of the brain.

(c) Selection representing the skin, cranium and meninges.

(d) All above selections together.

Figure 7: P-LSP projection of basins from watershed segmentation of the brain volume. Projecting the basins reduce the number of points from 1,114,112 to 65,279 on the final layout, preserving the outcome on separating the head parts.

4.3 Computational complexity and running time

The P-LSP computational complexity can be described as $O(C+P+I)$, where C is the complexity of creating the partitions, P is the complexity of projecting the control points, and I is the complexity of projecting each individual partition. The complexity of the bisecting k-means algorithm to create $\sqrt{n}$ clusters is $O(n\sqrt{n})$ [TSK05]. The complexity of projecting $n^{0.75}$ control points using LSP is $O(n\sqrt{n})$ – LSP is $O(n^2)$ when conjugated gradient is used to solve the system and A^TA is ill conditioned. Considering that bisecting k-means produce clusters of approximately equal sizes [SKK00], we have $\sqrt{n}$ partitions with $\sqrt{n}$ elements. Using LSP to project all these partitions is $O(n\sqrt{n})$. Therefore the complexity of P-LSP is $O(n\sqrt{n}+n\sqrt{n}+n\sqrt{n}) = O(n\sqrt{n})$.

In terms of running times, P-LSP is two or three orders of magnitude faster than the original LSP considering the data sets used here. Table 1 presents the size and the number of features of each data set, and the running times for creating the projections using the P-LSP and the LSP on a CPU Intel® Xeon® 2.33GHz, with 32GB of RAM memory. The source code of P-LSP to solve the linear systems and the other necessary steps, such as clustering, nearest neighbors queries, etc., is the same used by the basic LSP. These running times were taken by executing twice each technique for each data set, getting the average value of the obtained running times. Some values for the LSP technique are missing since our implementation was not able to calculate the projections in a reasonable amount of time on the aforementioned machine. This illustrates the degree of scalability now possible with the P-LSP over that of the LSP.

Table 1: Running time in seconds.

Data set	Size	#Features	LSP	P-LSP
Brain Image (Gabor)	64,435	30	16,868.994	33.064
Brain Image (Intensity)	64,435	14	3,523.094	26.504
Head Image (Gabor)	919,584	30	—	2,555.702
Papaya Volume	222,548	5	14,072.763	107.122
Head Volume	1,114,112	14	—	2,358.941
Head Volume (Watershed)	65,279	9	5,441.198	19.756
Vis 2008 contest	576,600	10	—	1,660.832

In the next section we analyze the precision of P-LSP and how little the layout has been affected by the approximations when compared to basic LSP.

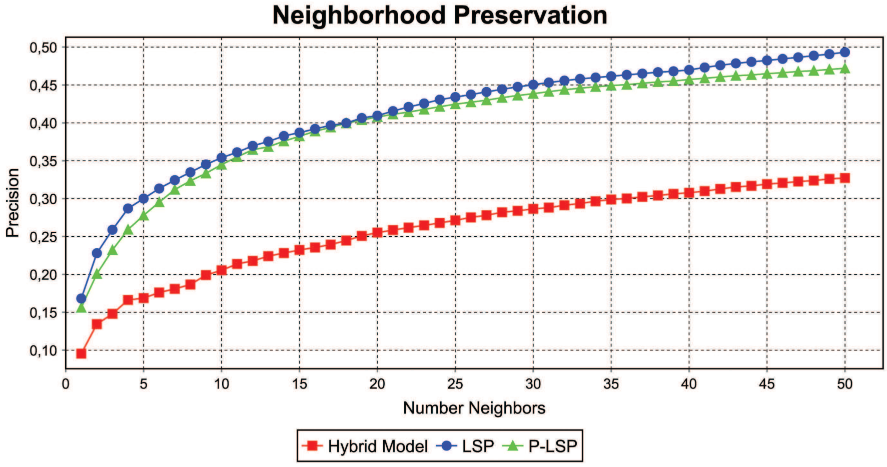


Figure 8: Neighborhood preservation comparison between P-LSP, LSP, and the Hybrid model.

4.4 Neighborhood preservation analysis

In order to evaluate the results produced by P-LSP, we reproduce the *Neighborhood Preservation (NP)* measure [PM08], comparing its results with the original LSP and also with the *Hybrid Model* presented in Section 2. The NP of one projected point is obtained by calculating its k-nearest neighbors on the $\mathbb{R}^2$ and verifying the percentage of preservation of the neighborhood defined on the original space $\mathbb{R}^m$. The NP of a projection is given as the average between the NP of all points belonging to it. The results for the Brain Tumor data set (intensity features) is given on Figure 8, with varying the number of nearest neighbors. Similar results are obtained for other data sets.

The attained results are satisfying in that, although somewhat less precise than the ones provided by LSP for small neighborhoods, they still remain excellent - and slightly better for larger neighborhoods. The slight fall in precision can be seen as a result of partitioning the data set and projecting the individual partitions. Since there is not a guarantee that the most similar elements of all instances belong to the same partition – an outcome of any clustering algorithm – some similar instances may belong to different partitions. Although the control points place the most similar partitions close to each other on the final layout, for small neighborhoods some misplacement is expected. This is a problem inherent to any dimension reduction technique. Nevertheless, the results are close to the ones produced by the LSP, and are more precise than the ones produced by the *Hybrid Model*, the other projection technique that took into account optimizations for increase in performance. We believe that the reasonably accuracy penalty is a small price to pay for the orders of magnitude performance increase.

5 Conclusions and Future Work

We have presented the Piecewise LSP, a high precision projection technique that is able to efficiently project high-dimensional samples to 2-D in such a way that their neighborhood relations are maintained. For example, points that are close together in a high-dimensional feature space will be close together in the 2-D visual representation.

The technique is a significant extension of LSP technique [PNML08] in that it is up to three orders of magnitude faster and keeps the quality of the produced layouts in terms of neighborhood preservation, enabling the exploration of large multi-variate data sets. One of the examples shown is a volume with more than a million voxels and 14 features per voxel. Importantly, this technique involves *all* features in the projection, and not just a subset.

Our strategy for projection based exploration in feature space coupled with visualizations in object space is unique in the sense that it can help users locate structures and patterns of interest in full high-dimensional feature volumes, without having to treat or interpret individual attributes or groups of attributes, or to pre-cluster based on attributes.

We have also presented an approach for exploring volume data sets by pre-partitioning them with the watershed algorithm and then projecting the basins using the P-LSP. Although the P-LSP is capable of projecting the multi-variate voxels of a complete volume data set, basin projection offers several advantages in terms of interactivity and clarity of the projection. This strategy supports faster interaction with the volume visualization and works beautifully with the idea of projecting groups of voxels to form groups of basins in visual space.

Additionally, the examples have shown support for users to converge, through iterations, to proper definition of feature sets and similarity measures, suggesting that this approach is also useful for supporting pre-setting of parameters for other automatic and mining tasks, such as segmentation and classification.

As future steps in the development we intend to expand the tool and the techniques to handle vector and tensor data. Due to the various partitions present in the methods disclosed here, it is also possible that adequate parallelization of the code will help improve interactivity further.

It is also suggested by the behavior or the proposed technique that a multi-level combination of basins when partitioning the volume before (or after) projection could possible help get rid of the artifacts caused by the watershed step.

Acknowledgements

Fruit images were provided by Embrapa - Brazilian Agricultural Research Corporation - São Carlos - Brazil.

References

[BBP07] Jorik Blaas, Charl P. Botha, and Frits H. Post. Interactive visualization of multi-field medical data using linked physical and feature-space views. In Ken Museth, Anders Ynnerman, and Torsten Möller, editors, *Proc. Eurographics / IEEE-VGTC EuroVis*, pages 123–130, 2007.

[BP08] C. P. Botha and F. H. Post. Hybrid scheduling in the devide dataflow visualisation environment. In H. Hauser, S. Strassburger, and H. Theisel, editors, *Simulation and Visualization*, pages 309–322. SCS Publishing House Erlangen, February 2008.

[CC00] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, second edition, 2000.

[Cha96] Matthew Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the IEEE Visualization 1996 (VIS'96)*, pages 127–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[DGH03] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.

[FR91] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[GRW$^+$00] D.L. Gresh, B.E. Rogowitz, R.L. Winslow, D.F. Scollan, and C.K. Yung. Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. In *Proceedings of IEEE Visualization 2000*, pages 489–492, 597, 2000.

[HO93] W. Higgins and E. Ojard. Interactive morphological watershed analysis for 3d medical images. *Computerized Medical Imaging and Graphics*, 17(4/5):387–395, 1993.

[JF91] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recogn.*, 24(12):1167–1186, 1991.

[JM04] F. Jourdan and G. Melançon. Multiscale hybrid mds. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference on (IV'04)*, pages 388–393, Washington, DC, USA, 2004. IEEE Computer Society.

[Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[MC04] A. Morrison and M. Chalmers. A pivot-based routine for improved parent-finding in hybrid MDS. *Information Visualization*, 3(2):109–122, 2004.

[MM96] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[MRC02] Alistair Morrison, Greg Ross, and Matthew Chalmers. A hybrid layout algorithm for subquadratic multidimensional scaling. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 152, Washington, DC, USA, 2002. IEEE Computer Society.

[PB95] N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE TFS*, 3(3):370–379, 1995.

[PM08] Fernando V. Paulovich and Rosane Minghim. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, 2008.

[PNML08] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008.

[Sam64] J. W. Sammon. A nonlinear mapping for data structure analysis. In *IEEE Transactions on Computers*, volume C-18, pages 401–409, May 1964.

[SB04] S. M. Savaresi and D. L. Boley. A comparative analysis on the bisecting k-means and the pddp clustering algorithms. *Intell. Data Anal.*, 8(4):345–362, 2004.

[SCO04] O. Sorkine and D. Cohen-Or. Least-squares meshes. In *Proceedings of Shape Modeling International*, pages 191–199. IEEE Computer Society Press, 2004.

[She94] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, University of California at Berkeley, August 1994.

[SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, pages 109–110, Boston, Massachusetts, USA, August 2000. ACM.

[TSK05] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[VS91] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[WN08] D. Whalen and M. L. Norman. Competition data set and description. In *2008 IEEE Visualization Design Contest*. http://vis.computer.org/VisWeek2008/vis/contests.html, 2008.