Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization

F.V. Paulovich[1], D.M. Eler[1], J. Poco [1,2], C.P. Botha[3,4], R. Minghim[1] and L.G. Nonato[1]

[1]ICMC/USP São Carlos/SP, Brazil, [2]University of Utah, USA
[3]Data Visualization, Delft University of Technology and [4]LKEB, Leiden University Medical Center, The Netherlands

Abstract

Multidimensional projection is emerging as an important visualization tool in applications involving the visual analysis of high-dimensional data. However, existing projection methods are either computationally expensive or not flexible enough to enable fully interactive data manipulation. That is, they do not support the feedback of user interaction into the projection process. A mechanism that dynamically adapts the projection based on direct user interaction would go a long way towards making the technique more useful with a large range of applications and data sets. In this paper we propose the Piecewise Laplacian-based Projection (PLP), a novel multidimensional projection technique, that, due to the local nature of its formulation, enables a versatile mechanism to interact with projected data and to allow interactive changes to dynamically alter the projection map, a unique capability of the technique. We exploit the flexibility provided by PLP in two interactive projection-based applications, one designed to organize pictures visually and another to build music playlists. These applications illustrate the usefulness of PLP in handling high-dimensional data in a flexible and highly visual way. We also compare PLP with the currently most promising projections in terms of precision and speed. The results show that PLP perform very well also according to these quality criteria.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques— H.5.0 [Information Interfaces and Presentation]: General—

1. Introduction

Much research has been done on creating mechanisms to handle multi-valued data. In visualization, most approaches rely on feature spaces to devise visual tools that assist in analyzing high-dimensional data. Among these techniques, multidimensional projection (MP) techniques have been playing an important part, even becoming an essential tool in recent visualization systems [DANS10, CWDH09], particularly due to the fact that they can, like no other method of visual analysis, handle large number of attributes and increasingly high numbers of data items successfully.

Despite their increasing acceptance, multidimensional projection techniques have disadvantages that restrict their use as fully interactive visual exploration tools capable of accompanying the analysis process to completion. For example, most multidimensional projection methods are global, that is, a single global transformation maps data instances from a high-dimensional space to the visual space. This global nature leads to difficulties maintaining locally the properties of grouping and group separation that the data may posses, thus preventing analysis of correlation between elements in a closer neighborhood to points of interest in the data set. In the large group of applications where the analysis starts as a global or overall interpretation but ends in the analysis of smaller groups up to individuals, the global nature of the available methods tends to hamper the user experience and prevent local adjustments to occur. Such local adjustments are necessary in order to incorporate user knowledge into the projection process. A few MP techniques provide mechanisms that could allow modifying the projection in accordance with user intervention, but the local ones fail, in terms of computational cost requirements, to be interactive and the global nature of others limits the type of changes that can be performed.

In this paper we propose a novel multidimensional projection technique, the *Piecewise Laplacian-based Projection (PLP)*. In contrast to most existing methods, PLP has a local character that renders it more versatile than other projec-

tion schemes in addressing the drawbacks discussed above. To complete the proposed solution, we present a mechanism for locally changing the projection, in accordance with user interaction, in such a way that the mapping itself adapts to user manipulations of the layout during visual exploration. This mechanism can be combined with the local nature of PLP so as to allow for drastic changes in the projection map, enabling the exploration and organization of the data in a flexible and dynamic way. The provided flexibility can be exploited in many applications, such as user-driven picture organization and music playlist construction, as described in Section 5.

Computation efficiency and accuracy are the other important properties of PLP. As we show in our results section (Section 4), compared to 10 techniques with 8 data sets varying from 1,500 to 250,000 points, PLP turns out to be quite effective, presenting accuracy comparable to the best existing techniques while still enabling interactive user intervention. The projection accuracy and flexibility in terms of layout dynamic adaptation render PLP an attractive projection technique for problems involving large multidimensional data visualizations and interactive exploration.

We can summarize the contributions presented in this paper as:

- PLP: A novel technique that relies on local rather than global maps to project high-dimensional data to visual space (Section 3). Accuracy is comparable to the best existing techniques, but, unlike most existing techniques, the projection is interactively steerable.
- *Neighborhoods from Visual Space*: A new mechanism to define neighborhoods in the high-dimensional space through manipulation of the visual space (Section 3.3), which allows for drastic changes in the local maps, in order to adapt them to the perception the user has of the data distribution.
- *Real Data Case Studies*: In Section 5 we show how our technique can be used to design a projection-based application to visually organize pictures and to create music playlists. To the best of our knowledge, this is the first time a multidimensional projection is explicitly employed to perform data organization interactively.

2. Related Work

Most projection methods derive from multidimensional scaling techniques (MDS). MDS methods perform embedding into a visual space by considering only distance measures (also called dissimilarities) between pairs of instances, rendering Cartesian coordinates for the original data instances unnecessary. When data is endowed with Cartesian coordinates, Euclidean distance can be used to generate the pairwise distances for MDS methods.

Projection and MDS methods vary greatly in terms of the mathematical foundation they rely on. Techniques based on spectral decomposition, for example, typically compute embedding coordinates for each data instance from eigenvectors of a double-centered transformation applied to the dissimilarity matrix (symmetric matrix containing the dissimilarity between each pair of data instances). Since the very first approach proposed by Torgeson [Tor65], much effort has been made to reduce the high computation costs associated with the eigendecomposition [BN03, KCH02, FL95]. Some spectral-based methods, such as Isomap [TdSL00], can also deal with distance measures other than Euclidean, thus accomplishing tasks such as "manifold unfolding". Although effective for dimensionality reduction purposes, these methods have a global nature and do not provide mechanisms for user intervention in the result, both shortcomings for many highly interactive applications with multilevel data analysis.

Spectral-based methods that make use of a more local methodology have also been described in the literature. An important representative is LLE [RS00]. Other good examples are Landmarks MDS [dST04] (L-MDS) and Pivot MDS [BP07]. LLE performs local linear fittings as the first processing step, accomplishing the final embedding through a global eigendecomposition approach. L-MDS and Pivot adopt an opposite scheme, first making use of an eigendecomposition to embed a subset of instances, mapping the remaining instances by an interpolation mechanism that relies on the eigenvectors computed in the first step. The eigendecomposition introduces a "global" component to those methods while still preventing interactive local changes. Therefore, methods based on spectral-decomposition can hardly be employed in interactive applications such as the ones proposed in this paper.

Nonlinear optimization methods rely on different schemes to find a minimum for an energy function, usually called the stress function. First proposed by Kruskal [Kru64], optimization methods tend to be computationally expensive, although reasonable performance can be reached by using multigrid-based numerical solvers, as shown by Bronstein et al. [BBKY06]. Following the idea of a subset of samples towards reducing computational cost, Pekalska et al. [PdRDK99] proposed an algorithm that first embeds a subset of samples using a gradient descent approach and then places the remaining instances using a global linear mapping.

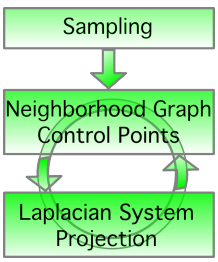
Force-based methods arose from the seminal work by Eades [Ead84], which makes an analogy between stress function minimization and mass-spring systems. The high computational cost of the algorithm proposed by Eades has been mitigated by Chalmers [Cha96] by making use of neighborhood structure and a subset of samples. Variants of Chalmers' algorithm with lower computational cost [MRC02, JM04, TMN03] and GPU implementation [IMO09, FT07] have also been proposed to speed-up

convergence and handle large data sets. While Chalmers' method can be seen as a local approach, computational times are still prohibitive for interactive applications.

Paulovich et al. [PNML08] proposed a technique called Least Squares Projection (LSP) that uses a force-based scheme to first position a subset of the samples, mapping the remaining instances through a Laplace-like operator. In contrast to our approach, Paulovich's method makes use of a global graph to build the multidimensional mapping, resulting in a large linear system. Moreover, LSP constraints the system through a least square procedure, rising computational cost considerably. Although LSP enables user intervention, its global nature and high computational cost means that the user can not freely change the projection layout. The same is true for the recent linear mapping PLMP [PSN10], which combines a force-based scheme to place representative instances in the visual space with a global linear mapping. The new PLP technique described in this paper also employs a Laplacian matrix to carry out the multidimensional projection. However, we make use of a dynamic mechanism to define neighborhood graphs from which we build a set of local Laplacian matrices to accomplish the mapping, thus avoiding global structures such as the ones present in LSP and PLMP.

3. The PLP Method

The PLP method is made up of three main components: sampling, neighborhood graph building, and Laplacian linear system solving, as illustrated on the right. Sampling refers to the selection of a small subset of instances. For each of these samples, a neighborhood graph and a set of control points are defined. The graph and control points associated with a given sample are used respectively to set and constrain the corresponding Laplacian systems. In other words, the overall idea is to associate a neighborhood graph and a set of control points to each given sample. Each graph gives rise to a Laplacian matrix that accomplishes the projection of the instances corresponding to the graph nodes. Control points are used to constrain the Laplacian system, thus steering the positioning of projected instances in the visual space. Projected instances can then be handled by the user so as to improve the grouping of similar instances. Neighborhood graphs and control points are dynamically updated during user intervention, thus modifying the Laplacian matrices and the resulting projections. Details on each step are presented in the following sections.



3.1. Sampling, Neighborhood Graphs and Control Points

Let $D = \{p_1, \ldots, p_n\}$ be a data set with instances in a d-dimensional space and $S = \{s_1, \ldots, s_m\}$ a subset of samples taken from D. The way one chooses the set S may vary depending on the application. For example, if the main goal is only to project D into the visual space then samples can be chosen using a clustering approach (see Section 4). Samples can also be provided by the user in order to drive the projection in accordance with a priori information. As we show in Section 5, the freedom to define samples can be exploited to design applications towards picture and playlist organization.

The samples S are used to split D into m subsets $D = D_1, \ldots, D_m$, where each subset D_i comprises the instances in D closer to s_i than to any other sample s_j, $j \neq i$. The D_i subsets can be computed in $O(mn)$ using the bisecting k-means [SKK00] technique. The number of samples m is chosen as $m = \sqrt{n}$ because this is an upper bound for the number of groups in a data set [PB95], thus most clusters should have a representative among the sample points. The neighborhood graph ND_i corresponding to D_i is defined as the *k-nearest neighbor graph (k-NNG)* connecting instances in D_i. Each node in ND_i represents an instance in D_i. Two nodes in ND_i are connected by an edge if at least one of them is among the k-nearest neighbors of the other. The parameter k is set to 10 in our implementation, as this value turned out to be a good compromise between computational cost (the larger k, the more costly to build the graph) and graph connectedness (the number of neighbors of each node). We noticed in our experiments that when k was small ($k < 5$) the graph D_i became weakly connected, impacting negatively on the result of the projection.

As mentioned before, each neighborhood graph ND_i gives rise to a Laplacian system that is used to project instances from D_i to the visual space. In order to ensure a unique solution for the projections, we have to impose constraints on the Laplacian system, which, in our case, are given through control points. The set of control points constraining the projection of D_i is defined by randomly picking out $\sqrt{n_i}$ instances from D_i, where n_i is the number of instances in D_i. As discussed in [PNML08], $\sqrt{n_i}$ randomly chosen control points yield a good balance between computational cost (control points have to be placed in the visual space using costly methods) and the quality of the final mapping. The main advantage of choosing the control points locally rather than globally is to ensure that each subset D_i has a number of control points proportional to its number of instances, a property difficult to attain with global selection. Details on how to build the Laplacian matrices and their corresponding constraints will be discussed in the next subsection.

3.2. The Laplacian system

The Laplacian-based projection mechanism 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 visual domain. In more mathematical terms, let p_i be an instance in D_i and $Viz(p_i) = \{p_{i_1}, \ldots, p_{i_k}\}$ be the set of nodes connected to p_i in ND_i. Let also $(x_{p_{i_j}}, y_{p_{i_j}})$ be the coordinates of each element $p_{i_j} \in Viz(p_i)$ when mapped to the visual space $\mathbb{R}^2$. Assuming the convex combination hypothesis, the two-dimensional coordinates of p_i can be written as:

$$\overline{p}_i = (x_{p_i}, y_{p_i}) = \sum_{p_{i_j} \in Viz(p_i)} \alpha_{ij} (x_{p_{i_j}}, y_{p_{i_j}}) \qquad (1)$$

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

Each element in D_i 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) given by:

$$L_{ij} = \begin{cases} 1, & \text{if } i = j, \\ -\alpha_{ij}/\alpha_i^*, & \text{if } i \neq j \text{ and } p_{i_j} \in Viz(p_i) \\ 0, & \text{otherwise.} \end{cases}$$

where $\alpha_i^* = \sum_{p_{i_j} \in Viz(p_i)} \alpha_{ij}$. The weight α_{ij} can be set as the inverse of the distance between p_i and p_j or simply equal to one, giving rise to the so called combinatorial Laplacian. Although the weighted graph Laplacian can also be used (see [PNM06]) we choose the combinatorial Laplacian in our implementation, since it produces good visual results while still being numerically more robust. Moreover, as we show in Section 5, the neighborhood graphs are dynamically updated in the interactive applications. Designing a consistent heuristic for assigning weights to new edges created during user interaction is difficult.

It can be shown that if ND_i has only one connected component then the rank of L is $n_i - 1$. Thereby, assuming ND_i connected, the linear systems (2) admit a non-trivial solution. The lack of geometric information in (2) can lead to solutions that are difficult to interpret and analyze. PLP deals with this problem by constraining the systems with geometrical information provided by the *control points*. The rationale is to project the control points associated with each D_i by a MDS method then use their x and y coordinates in the visual space to constrain the Laplacian systems. In our implementation we use the force-based scheme [TMN03] to project the control points. Since the number of control points is just a fraction of the number of instances in D_i, the high computational cost of the force-based scheme is not an issue. Moreover, we use a penalty method to constraint the system [XZCOX09], speeding up the underlying numerical manipulation.

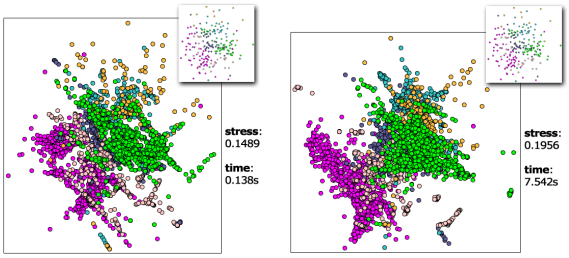


Figure 1: *Left: Control points from all subsets D_i are simultaneously embedded in the visual space and their Cartesian coordinates used to constrain the Laplacian systems of PLP. Right: Projection generated from a single global Laplacian system constrained by the same control points used on the left. Numbers show computational times and the stress function. Results appear similar, but PLP (on the left) is more accurate (lower stress) and faster.*

An advantage of using control points to constrain the systems (2) is that we can preserve coherence when mapping the subsets D_i. In other words, if we project each D_i independently, no guarantee can be given towards ensuring that neighbor subsets will be mapped close to each other in the visual space. However, by handling the control points properly one can attain a global relationship among groups without losing the local processing benefit of PLP. More specifically, the global relation can be built as follows: Let C_i be a set of control points chosen from a subset D_i. Consider now the set $C = C_1, \ldots, C_m$ comprising the union of control points from all subsets D_i, $i = 1, \ldots, m$. The set C can be seen as a new data set containing a fraction of the instances from D. If the set of control points C is embedded in the visual space using the force-based scheme, the obtained x and y Cartesian coordinates of the control points originating from a particular subset D_i will be in unison with the control points of other subset D_j, thus reintroducing the global correspondence among the subsets lost during the partition stage. This global control point mapping will keep similar groups close to each other, putting apart dissimilar subsets.

Figure 1 shows a comparison of projecting a data set using the PLP with all control points embedded simultaneously (left) and the layout produced by using the same control points but a single Laplace system (right) – the approach employed by LSP. Both approaches produce similar results, however, PLP turns out to be more accurate (lower stress) and computationally faster.

3.3. Handling Control Points and Neighborhood Graphs

As shown above, PLP can present a global behavior if control points from all subsets D_i are simultaneously embedded

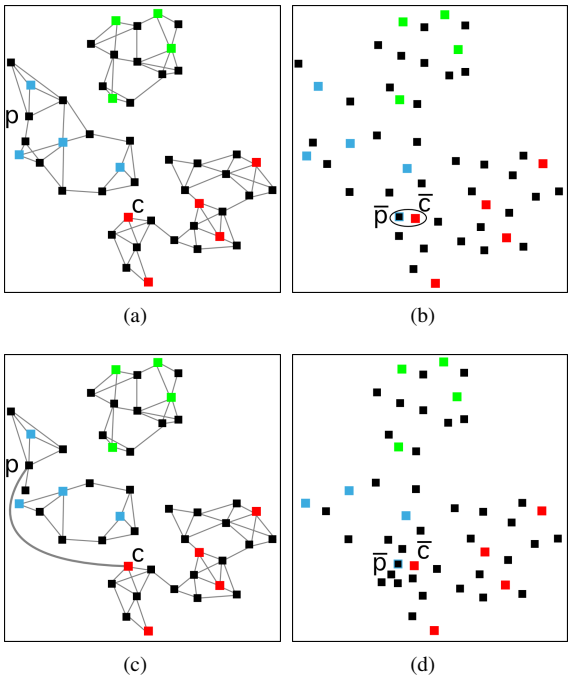


Figure 2: *Given a projection (a) the user can drag and reposition projected instances (b). Neighborhood graphs are updated to reflect the user defined neighborhood relationship (c), thus modifying the Laplacian matrices and the projection (d).*

into the visual space using a global MDS approach. However, the local nature of PLP allows for modifying the projection in accordance with user intervention.

The rationale is to carry out local modifications by changing the neighborhood graphs and control points. More specifically, suppose a data set D has been projected using the procedure described in the previous section (Figure 2(a)). The user can interact with projected instances, picking out a particular instance $\overline{p}$ (overline will be used to denote instances in the visual space) in the visual space and dragging it to a new position, as illustrated in Figure 2(b).

Let $\overline{c} \in C$ be the control point closer to $\overline{p}$ and D_c, D_p be the subsets containing c and p respectively. There are two cases to be considered, either $p \in D_c$ or $p \notin D_c$. If p is in D_c then $ND_p = ND_c$ and the only change we carry out is to add an edge connecting p to c in neighborhood graph, which induces a change in the Laplacian matrix associated to ND_c. If $p \notin D_c$ then we not only add an edge connecting p and c but also move p and its neighbors from ND_p to ND_c, as illustrated in Figure 2(c). If ND_p becomes disconnected due to the removal of p then we add new edges between the disconnected parts. The new edges connect the 10 nearest samples in both disconnected parts. This heuristic violates the KNN property, but ensures full rank for the Laplace system.

In short, the graph updates are just enforcing instances to become neighbors in the neighborhood graph, even though they are far from each other in the original high-dimensional space. Figure 2(d) illustrates the resulting projection after updating neighborhood graphs with the new user-driven neighborhood relationship.

We also consider the possibility of just separating groups of instances. If the user drags $\overline{p}$ to a position in the visual space where its distance to $\overline{c}$ is larger than a threshold then we consider the user wishes to create a new subset D_i. In this case, we run the procedure described in 3.1 in the subset D_p, using p as a new sample.

The novelty in the process described above is to drive changes in the neighborhood graph by interacting in the visual space. This mechanisms allows the user to interact with projected data quite freely, visually regrouping and segmenting the data, as we show in the following.

4. Results and Comparisons

In this section we present the results of applying PLP to project several distinct data sets. We also provide a comprehensive set of comparisons to assess the accuracy and speed of PLP. All the results were generated in an Intel® Core™ i7 CPU 920 2.66GHz, with an NVIDIA® Quadro FX 3800 video card and 8GB of RAM. PLP is implemented in Java, as is the numerical solver – we use the Cholesky factorization available on Java Colt Project (`http://acs.lbl.gov/~hoschek/colt/`). We are using Cholesky because two linear systems have to be solved and the factorization of one system can be used in the solution of the other, resulting in a performance gain.

We start by showing how the PLP handles the unfolding problem, as presented in Figure 3. Figure 3(a) shows the so called *Swiss Roll* data set and Figures 3(b) and 3(c) present the resulting PLP projection when a force-based scheme [TMN03] and the Isomap [TdSL00] are respectively employed to embed control points in the visual space. Notice that PLP is able to unfold the surface when Isomap is employed to project control points (Figure 3(c)). For the sake of comparison, Figure 3(d) shows the result of projecting the Swiss Roll data set with the PLMP [PSN10]. PLMP also requires as first step the projection of a subset of representatives, and in Figure 3(d) we employed Isomap to embed the required representatives. Due to its global nature, PLMP could not unfold the Swiss Roll, even though geodesic distances have been used to project representatives.

Although PLP has been tailored mainly to be an interactive projection tool, accuracy and computational time are both competitive. We have employed eight distinct data sets in our experiments, some of them synthetic, allowing comparison between PLP's performance and other available techniques employing data sets that vary with enough variation size and data dimensionality. The data set *WDBC* is a breast cancer data set obtained from digitized images of

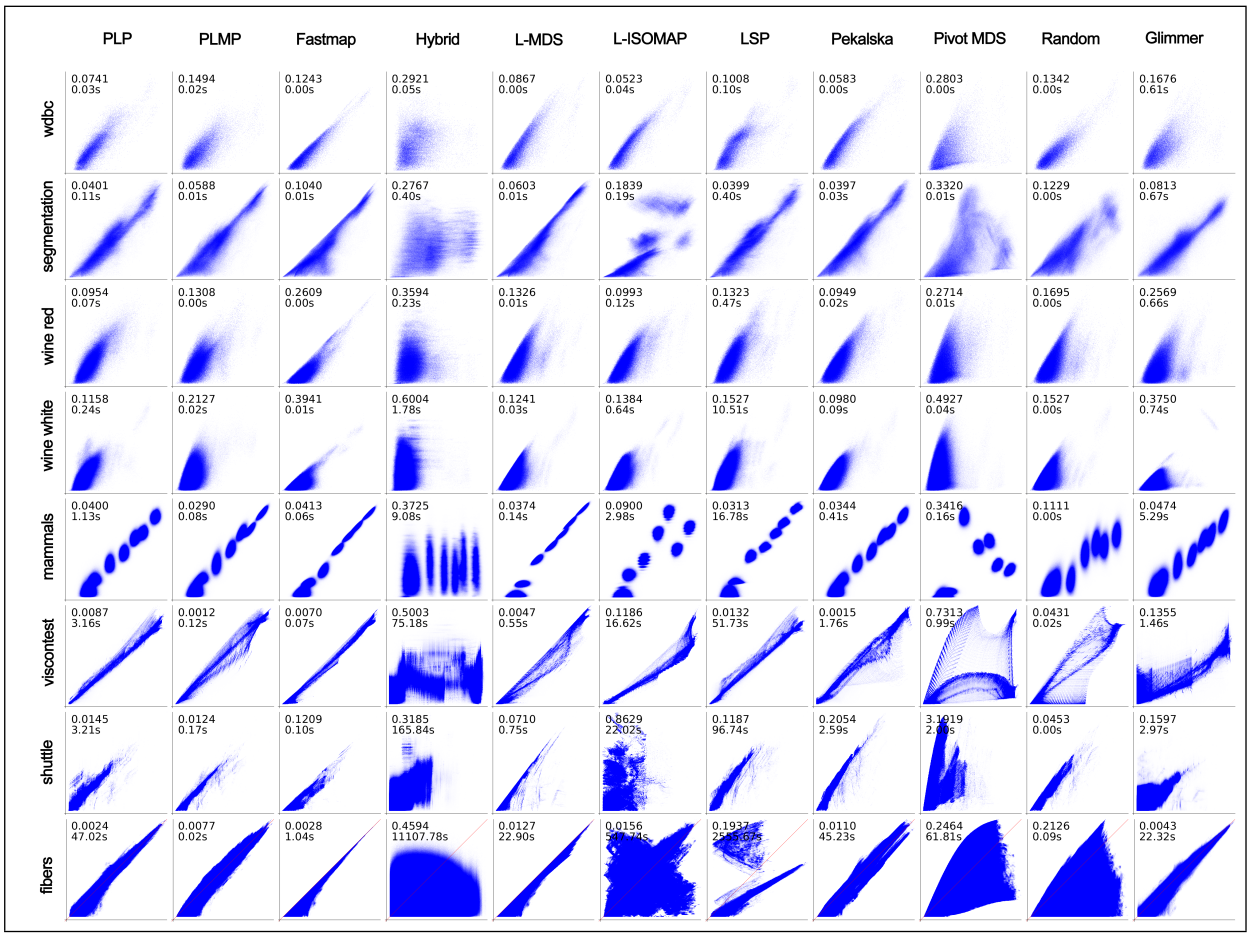
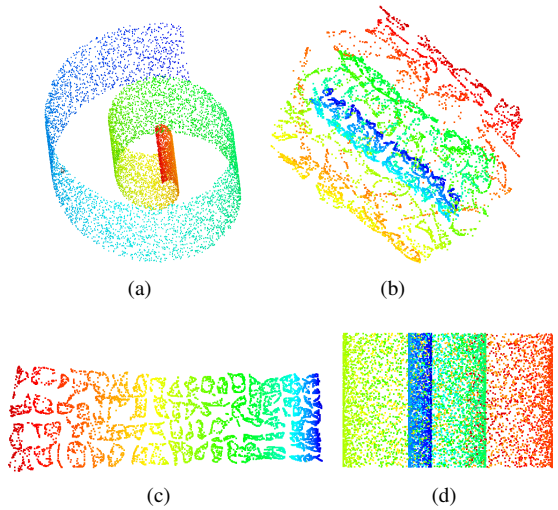


Figure 4: `original-distance` × `projected-distance` *scatter plots. From left to right PLP, PLMP [PSN10], Fastmap [FL95], Hybrid [JM04], Landmarks MDS [dST04], L-Isomap [dST03], LSP [PNML08], Pekalska [PdRDK99], Pivot-MDS [BP07], Random Projection [Ach03] and Glimmer [IMO09]. Top-left numbers are respectively the normalized stress and computational time in seconds.*

breast masses. It has 539 instances in thirty-dimensional space that have been classified into two distinct groups: malignant and benign cancer. *Wine-red* (1,599 instances with 11 dimensions) and *Wine-white* (4,989 instances with 11 dimensions) are related to red and white variants of the Portuguese "Vinho Verde" wine. The *Segmentation* (2,100 instances in 19 dimensions) data set is composed of features of 3×3 regions of a set of 7 outdoor manually segmented images. *Shuttle* (43,500 instances with 9 dimensions) is composed by log information instances split into 7 different classes. The *Mammals* (10,000 with 72 dimensions) is an artificially generated data set representing different features of mammals belonging to four distinct classes (dogs, cats, horses, and giraffes). *Viscontest* (30,000 instances with 10 dimensions) corresponds to a sample of time step 99 of a data set containing information from a simulation of an ionization front instability propagation during the formation of a galaxy. The *Viscontest* data set was obtained from the *IEEE Visualization 2008 Contest data set* [WN08] and the remaining ones were recovered from the *UCI Machine Learning Repository* [AN07]. Finally, *Fibers* (250,000 instances with 30 dimensions) was obtained from the *2009 Pittsburgh Brain Competition (PBC) – Brain Connectivity Challenge* (`http://pbc.lrdc.pitt.edu/`).

Figure 4 shows a comparison of PLP with ten other techniques, including two state-of-the-art methods in terms of projection methods for visualization, namely, PLMP [PSN10] and Glimmer [IMO09]. These ten methods have been chosen because they are shown to be, amongst the studied methods, the ones that present the best trade-off between accuracy and running times. The `original-distance` × `projected-distance` scatter plots clearly show that PLP outperforms most of the techniques, since it results in an almost 45^o diagonal layout, meaning that orig-



(a) (b) (c) (d)

Figure 3: *Unfolding problem. (a) The Swiss Roll data set. (b) PLP using force-scheme and Euclidean distances to embed control points. (c) PLP using Isomap to embed control points. (d) PLMP using Isomap to embed representative instances.*

inal distances are well preserved in the resulting projection. Numbers at the top-left of each plot correspond to computational time in seconds and the normalized stress given by $\frac{\sum_{ij}(d_{ij}-\overline{d}_{ij})^2}{\sum_{ij} d_{ij}^2}$ (d and $\overline{d}$ are the distance between instances p_i and p_j in the original and visual space).

Besides demonstrating the effectiveness of PLP, the `original-distance × projected-distance` scatter plots also bring out issues that are not apparent from error measures such as the normalized stress. Note for example in Figure 4 that L-Isomap presents a low stress value when projecting the Fibers data set. However, one can easily notice from the scatter plot that distances are not consistently preserved by L-Isomap. The opposite is also true, the scatter plot generated by projecting the Shuttle data set using Pekalska technique results in a high value of stress, but distances do not deviate significantly from the ideal 45^o line.

Figures 5(a) and 5(b) depict boxplots generated from normalized stress and computational times, shown on top-left of each plot in Figure 4. It is evident from Figure 5(a) that PLP is one of the most accurate methods, figuring among the best methods described in the literature, such as Pekalska and L-MDS. Regarding computational times, Figure 5(b) shows that PLP is at least an order of magnitude slower than PLMP, Fastmap, L-MDS and Random.

Regardless of the loss in processing time against other similarly accurate methods, the values are motivating enough for developing interactive applications of large data sets. In terms of the interaction capability, and particularly on the dynamic of changes, PLMP is the only method that

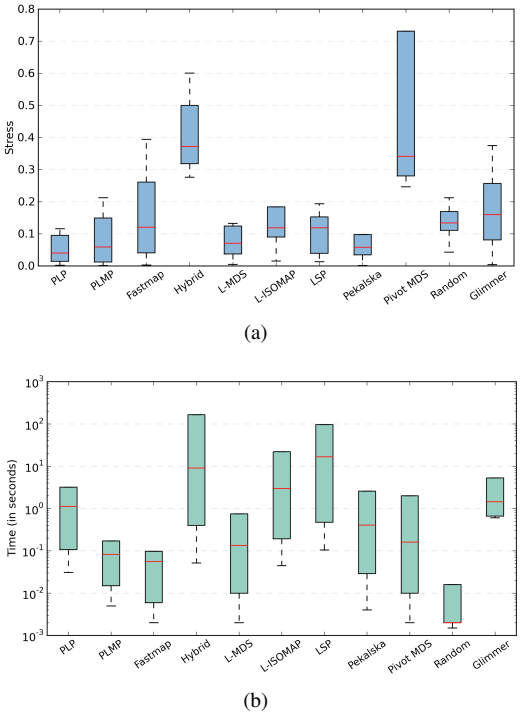


Figure 5: *Stress and computational times boxplots.*

enables interactive modification to the projection map while being faster than PLP.

In the issue of dynamic changes lies the greatest advantage of PLP. The intrinsic local nature of PLP supports drastic changes in the projection to be fed back into the mapping process so as to reconstitute mapping to be in the form defined by the user. This particular feature is not shared by PLMP, as Figure 6 clearly shows. Notice from Figure 6(b) that PLP was able to preserve the groups defined by user handling of the control points interactively. Due to their global nature, PLMP and LSP could not strictly follow the displacement of the control points (see Figures 6(c) and 6(d)). Therefore, PLP is the only technique able to follow the user-driven layout with acceptable computational cost.

It is important to point out that computational times shown in Figures 4 and 5(b) include the time spent to compute the groups D_i, the neighborhood graphs, the placement of control points, the Laplace matrices, Cholesky factorization, and the projection itself. During user interaction, though, updates take place only locally, thus demanding just small changes in the underlying structures. In fact, localized changes can be accomplished very quickly. In the tests we have carried out, PLP took around 90ms to update each subset D_i that had changed after user interaction (considering a projection with $250,000$ instances). This rate renders PLP as a fully interac-

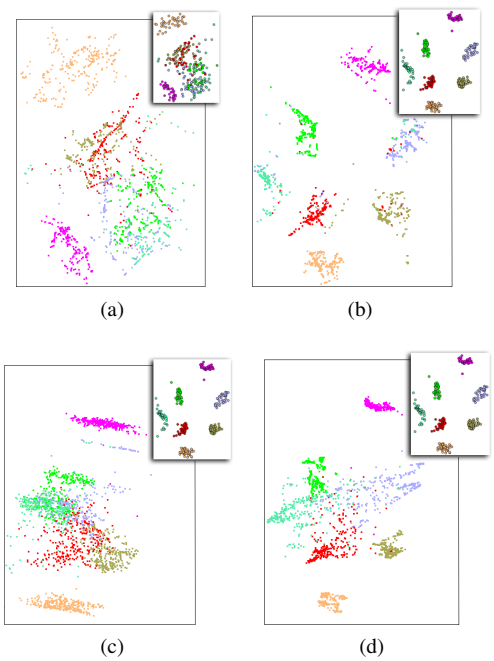


Figure 6: *Changing the projection map by repositioning control points. Top right window shows the position of the control points. (a) Projection generated by PLP considering using control points embedded by a force-based scheme. (b) PLP's result after repositioning control points in accordance with its classes. Groups were preserved in the final projection. (c) and (d) Projections generated by PLMP and LSP respectively using the same control points as in (b). Group separation is not preserved in the final projection.*

tive high dimensional data exploration and organization tool, as we show in the applications described below.

5. Applications

Making it possible for a projection to be guided by user knowledge through flexible interaction with the projected data is useful functionality that can be exploited in many data visualization problems. To demonstrate this, we have integrated the PLP framework in two visual data exploration and organization applications that are based on similarity. One application is a system to organize sets of images, and the other is a system to support the creation of music playlists. The idea is to ask the user to provide a set of "seeds" (the samples discussed in Section 3) from which the subsets D_i and the control points are defined. Control points are embedded in the visual space using the force-based scheme and they can be manipulated to improve grouping of similar instances. Finally, the whole data set is projected onto the visual space using the Laplacian maps.

Image Grouping We use the *Caltech database* in our image grouping experiments. The data base contains 3,812 colored images organized in 6 unbalanced classes: airplanes (1074 images), buildings (750 images), cars (526 images), faces (450 images), leaves (186 images) and motorbikes (826 images) [FPZ03]. We employ the *bag-of-visual features (BoVF)* [YJHN07] approach to compute the image features. The BoVF was set with a "vocabulary" of 150 features. The vocabulary was built from 50,000 keypoints obtained with the Harris-Laplace point detector and dense sampling [MTS*05]. Features from each keypoint were then extracted with the SIFT method [Low04].

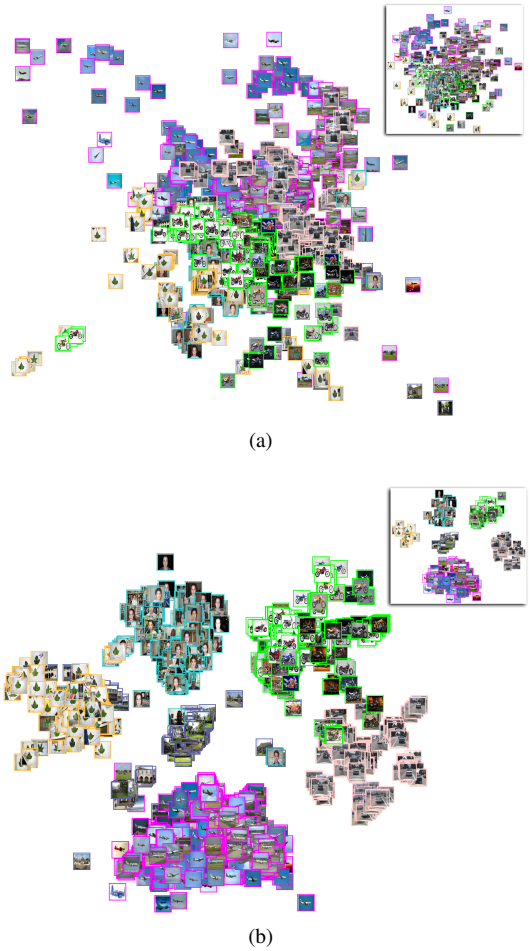


Figure 7: *(a) Image collection projected with control points positioned using the force-based scheme. The features are not good enough to separate the distinct group of images. The color of the borders indicates the class each image belongs to. (b) A new projection is defined repositioning control points, resulting in a good separation of image groups. The top right window shows the position of control points.*

Notice from Figure 7 that before user manipulation the projection of the pictures overlaps different classes. The user can modify the projection map to improve or create new groups of similar instances in the projection (see Fig-

ure 7(b)). We have implemented a selection tool that allows to pick out and displace a set of projected instances simultaneously, making the re-arrangement of control points a simple task. In fact, fewer than thirty interactions were need to organize the pictures as presented in Figure 7(b). In this application control points are directly handled, but any projected instance can be moved freely (the control points closer to the moved instances are also displaced) to change projection maps and thus the final layout.

Playlist Construction The image grouping system presented above can be modified to generate music playlists. We use a database with 3,857 music tracks and *JAudio Tool* [FM06] is used to extract low-level features from `mp3` files, such as beat points, statistical summaries, and so on, resulting in vectors with 78 dimensions.

Figure 8 shows screen shots of the system. The user starts by selecting a few music tracks (seeds) from a list containing the names of the artists and music titles (top-left screen in Figure 8). The system uses the user selected music tracks as samples. Some samples are also automatically computed to represent music tracks that are very different from the ones provided by the user. The idea is to create groups that represent music tracks the user is not interested in, which work as repositories for musics that should not be in the playlists.

From the user-defined as well as automatically computed seeds, the system compute the groups and the control points associated to each group. The control points are embedded in the visual space by force-based scheme and some of them are displayed in the system main window (Figure 8(a)). The whole set of control points is not displayed to avoid cluttering. The user can then interact with the control points, dragging some music tracks around to change the projection and thus the arrangement of musics that make up the initial playlists (Figure 8(b)). Finally, the whole data set is projected, being the playlists defined by the instances belonging to the neighborhood graphs containing the user-defined seeds (Figure 8(c)).

An advantage of the system prototype described above is that multiple playlists can be built simultaneously, functionality that is not available in any commercial tool available, making the creation of playlists a less time consuming task (see the accompanying video). To the best of our knowledge, this is the first time a multidimensional projection technique has been employed as a user driven data organization tool.

6. Discussion and Limitation

The comparisons presented in Section 4 clearly show that PLP is an efficient projection scheme, surpassing, in certain requisites such as accuracy and interactivity, the state-of-art methods. Its good performance is a consequence of combining the piecewise Laplacian mappings with the new mechanism to update neighborhood graphs, which enable local modification of projections in a cost effective way.

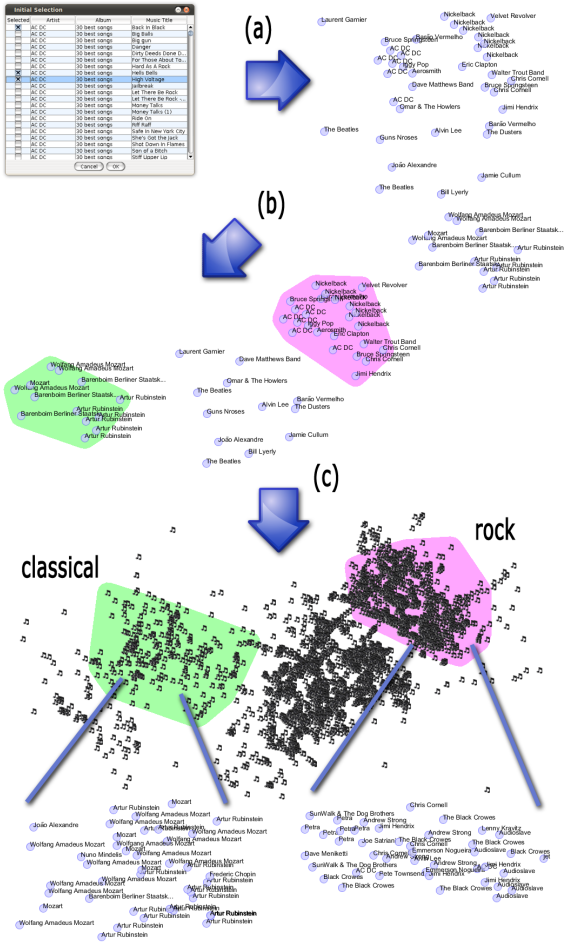


Figure 8: *Playlist maker: The user starts selecting sample music tracks to seed the playlists (top-left). Neighborhood graphs and control points are computed from the sample tracks. Control points are embedded in the visual space and some of them are displayed (a). User interacts with control points grouping the most similar ones (b). Finally, the whole data set is projected and playlists are created from the neighborhood graphs containing the user-defined samples.*

The `original-distance` × `projected-distance` scatter plots provide a convincing visual evidence of the accuracy of PLP while also supporting the assessment of the accuracy of other projection methods. Another important characteristic of PLP is its simplicity, essentially requiring the construction of neighborhood graphs, Laplacian matrices and a numerical linear solver.

The idea of dynamically updating neighborhood graphs in accordance with user intervention not only allows updating the underlying structures efficiently but also provides a natural mechanism to define groups. The nodes of each connected graph resulting from user interaction corresponds, in-

deed, to instances belonging to the same group. We have exploited that property to create music playlists.

A limitation of PLP is that the computational cost to update neighborhood graphs during interaction depends on the number of instances handled simultaneously. That might be a problem if the user selects a large number of instance to be repositioned simultaneously, as most of the neighborhood graphs have to be updated, thus resulting in intensive computation that could impact interactivity. To avoid this problem one can limit the number of instances to be selected simultaneously, however, this might lead to an increasing number of user interventions during the data organization process. Neighborhood graphs containing too few or too many instances may show up after a large number of interactions, which can affect the performance of the system. A possible solution is to keep track of the number of elements in each graph, merging small graphs and splitting large ones.

7. Conclusion and Future Work

In this work we proposed a novel projection technique called Piecewise Laplacian-based Projection, or PLP, which is shown to be accurate and cost effective in applications demanding user intervention. The evaluation we provided shows that PLP outperforms existing projection methods with respect to stress minimization as well as interactive data exploration and organization. Moreover, the potential of using PLP to interactively analyze multidimensional data sets opens new possibilities for applications which have not been addressed until now, due to the poor performance of existing projection methods in interactive applications, or due to global definitions that prevent user participation in important decision points. Flexibility, effectiveness, and ease of implementation, though, render PLP an attractive projection method for a large variety of applications.

We are investigating the applicability of PLP as an interactive tool in streaming data. The possibility of interactively changing the projection combined with streaming the projection should result in an powerful tool for applications such as remote sensing and surveillance.

Acknowledgments

The authors acknowledge the financial support of the Brazilian financial agencies CNPq and FAPESP.

References

[Ach03] ACHLIOPTAS D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci. 66*, 4 (2003), 671–687.

[AN07] ASUNCION A., NEWMAN D. J.: UCI machine learning repository, 2007.

[BBKY06] BRONSTEIN M. M., BRONSTEIN A. M., KIMMEL R., YAVNEH I.: Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications 13* (2006), 149–171.

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. 15*, 6 (2003), 1373–1396.

[BP07] BRANDES U., PICH C.: Eigensolver methods for progressive multidimensional scaling of large data. In *LNCS*, Kaufmann M., Wagner D., (Eds.), vol. 4372. 2007, pp. 42–53.

[Cha96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *IEEE Visualization* (1996), pp. 127–ff.

[CWDH09] CHEN Y., WANG L., DONG M., HUA J.: Exemplar-based visualization of large document corpus. *IEEE Trans. Vis. Comput. Graph. 15* (2009), 1161–1168.

[DANS10] DANIELS J., ANDERSON E. W., NONATO L. G., SILVA C. T.: Interactive vector field feature identification. *IEEE Trans. Vis. Comput. Graph. 16* (2010), 1560–1568.

[dST03] DE SILVA V., TENENBAUM J. B.: Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15* (2003), MIT Press, pp. 705–712.

[dST04] DE SILVA V., TENENBAUM J. B.: *Sparse multidimensional scaling using landmark points*. Tech. rep., Stanford, 2004.

[Ead84] EADES P. A.: A heuristic for graph drawing. In *Congressus Numerantium* (1984), vol. 42, pp. 149–160.

[FL95] FALOUTSOS C., LIN K.: FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases. In *ACM SIGMOD* (1995), pp. 163–174.

[FM06] FUINAGA I., MCENNIS D.: On-demand metadata extraction network (OMEN). In *ACM/IEEE-CS Joint Conf. on Digital Libraries* (2006), pp. 346–346.

[FPZ03] FERGUS R., PERONA P., ZISSERMAN A.: Object class recognition by unsupervised scale-invariant learning. In *CVPR* (2003), vol. 2, pp. 264–271.

[FT07] FRISHMAN Y., TAL A.: Multi-level graph layout on the GPU. *IEEE Trans. Vis. Comput. Graph. 13* (2007), 1310–1319.

[IMO09] INGRAM S., MUNZNER T., OLANO M.: Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Vis. Comp. Graph. 15*, 2 (2009), 249–261.

[JM04] JOURDAN F., MELANÇON G.: Multiscale hybrid MDS. In *Information Visualisation* (2004), pp. 388–393.

[KCH02] KOREN Y., CARMEL L., HAREL D.: ACE: A fast multiscale eigenvectors computation for drawing huge graphs. In *IEEE Information Visualization* (2002), pp. 137–144.

[Kru64] KRUSKAL J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29* (1964), 115–129.

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[MRC02] MORRISON A., ROSS G., CHALMERS M.: A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *IEEE Information Visualization* (2002), pp. 152–158.

[MTS*05] MIKOLAJCZYK K., TUYTELAARS T., SCHMID C., ZISSERMAN A., MATAS J., SCHAFFALITZKY F., KADIR T., GOOL L. V.: A comparison of affine region detectors. *International Journal of Computer Vision 65*, 1-2 (November 2005), 43–72.

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

[PdRDK99] PEKALSKA E., DE RIDDER D., DUIN R. P. W., KRAAIJVELD M. A.: A new method of generalizing Sammon mapping with application to algorithm speed-up. In *Annual Conf. Advanced School for Comput. Imag.* (1999), Boasson M., Kaandorp J. A., Tonino J. F. M., Vosselman M. G., (Eds.), pp. 221–228.

[PNM06] PAULOVICH F. V., NONATO L. G., MINGHIM R.: Visual mapping of text collections through a fast high precision projection technique. In *International Conference on Information Visualization* (2006), pp. 282–290.

[PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Trans. Visual. Comp. Graph. 14*, 3 (2008), 564–575.

[PSN10] PAULOVICH F. V., SILVA C. T., NONATO L. G.: Two-phase mapping for projecting massive data sets. *IEEE Trans. on Vis. Comp. Graph. 16*, 6 (2010), 1281–1290.

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (December 2000), 2323–2326.

[SKK00] STEINBACH M., KARYPIS G., KUMAR V.: A comparison of document clustering techniques. In *Workshop on Text Mining, ACM SIGKDD International Conference on Data Mining* (2000), pp. 109–110.

[TdSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (December 2000), 2319–2323.

[TMN03] TEJADA E., MINGHIM R., NONATO L. G.: On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization 2*, 4 (2003), 218–231.

[Tor65] TORGESON W. S.: Multidimensional scaling of similarity. *Psychometrika 30* (1965), 379–393.

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

[XZCOX09] XU K., ZHANG H., COHEN-OR D., XIONG Y.: Dynamic harmonic fields for surface processing. *Computer Graphics 33*, 3 (2009), 391–398.

[YJHN07] YANG J., JIANG Y.-G., HAUPTMANN A. G., NGO C.-W.: Evaluating bag-of-visual-words representations in scene classification. In *International Workshop on Multimedia Information Retrieval* (2007), pp. 197–206.