Local Affine Multidimensional Projection

Paulo Joia, Fernando V. Paulovich, Danilo Coimbra, José Alberto Cuminato, Luis Gustavo Nonato

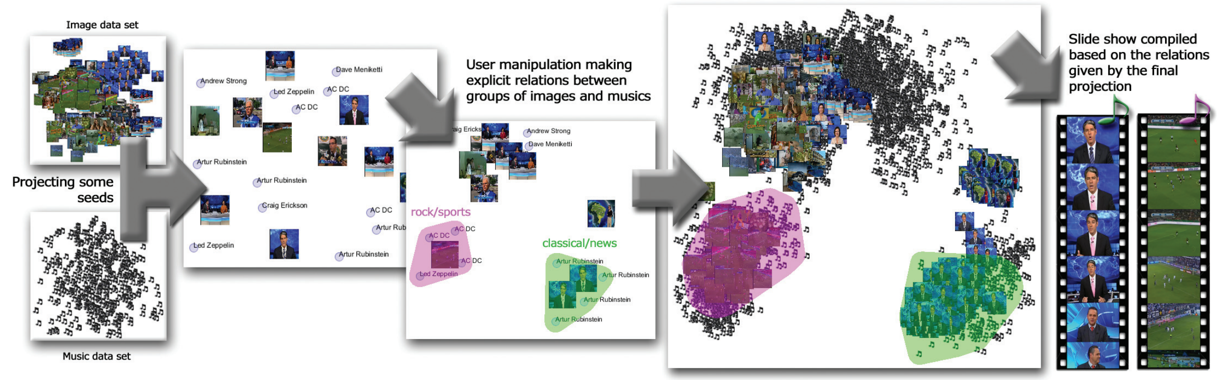


Fig. 1. Using projections to correlate different data sets that do not have explicit relation among instances. An initial projection is created using a few instances of each data set (music and images). Then, the relation amongst selected instances is defined by grouping images and music in the visual space, creating an explicit correlation. Considering this initial manipulation, the projections of the entire data sets are accomplished and the correspondence is settled. Finally, the lists of corresponding elements are used to produce slide shows where the images and related music are played in a synchronized manner.

Abstract— Multidimensional projection techniques have experienced many improvements lately, mainly regarding computational times and accuracy. However, existing methods do not yet provide flexible enough mechanisms for visualization-oriented fully interactive applications. This work presents a new multidimensional projection technique designed to be more flexible and versatile than other methods. This novel approach, called Local Affine Multidimensional Projection (LAMP), relies on orthogonal mapping theory to build accurate local transformations that can be dynamically modified according to user knowledge. The accuracy, flexibility and computational efficiency of LAMP is confirmed by a comprehensive set of comparisons. LAMP's versatility is exploited in an application which seeks to correlate data that, in principle, has no connection as well as in visual exploration of textual documents.

Index Terms—Multidimensional Projection, High Dimensional Data, Visual Data Mining.

1 Introduction

Multidimensional projection (MP) is increasingly becoming a fundamental tool in most visualization systems. Such a rise in popularity is motivated by the capability that MP methods have to handle large data sets with instances embedded in high dimensional attribute spaces. Vector field analysis [9], visual text mining [7, 24], and word cloud formation [8] are just a few examples of visualization-oriented applications where multidimensional projection has successfully been employed.

The effective use of MP methods in visualization has only become possible due to the advances these methods has experienced recently, which has pushed computational times down while keeping high degree of accuracy [26]. Despite the progress, MP methods still bear weaknesses that impair their use as fully interactive visual exploration tools. For example, most MP methods make use of a single global mapping to project data instances from a high-dimensional space to the visual space. This global nature hampers the user experience and prevents local adjustments to occur, that is, local changes affect the projection as a whole. MP methods based on local transformations also have deficiencies, as they either present high computational cost or do not provide mechanisms flexible and robust enough to permit the user to freely intervene in the projection. One of the main reasons for such a lack of flexibility is that existing local methods accomplish the multidimensional projection based on a subset of samples a priori positioned in the visual space and the number of samples required is typically high. Therefore, many instances have to be manipulated in order to properly modify the projection, thus making the interaction process tedious and time consuming in many cases.

This work presents a novel multidimensional projection technique called *Local Affine Multidimensional Projection (LAMP)*, which is endowed with unique properties that make it effective in addressing the drawbacks discussed above. LAMP relies on a mathematical formulation derived from orthogonal mapping theory, what ensures robustness and accuracy to the process. Moreover, the mathematical formulation can be tuned to make LAMP a local method that requires a reduced number of samples to build the mappings. Therefore, just a few interactions are need to incorporate user knowledge into the projection process, increasing flexibility. The local nature combined with the flexible interactive mechanism allow for dynamic exploration and organization of data, a trait that can be exploited in many applications. In particular, such a flexibility is brought to bear in a user-driven data correlation application and in visual exploration of textual documents, which are detailed in Section 5.

In summary, the main contributions of this work are:

- *LAMP*: A novel multidimensional projection technique that derives from orthogonal mapping theory (Section 3). LAMP can be tuned to be global as well as local, projecting data to the visual space in an accurate manner (see Section 4). The capability to deal with a quite reduce number of control points renders LAMP

- *P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato and L. G. Nonato are with Universidade de São Paulo, Brazil, Email: {pjoia,paulovic,danilobc, jacumina,gnonato}@icmc.usp.br.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

suitable for interactive applications.

- *Data Correlation*: The LAMP's flexibility and effectiveness is exploited in a new visualization-based data correlation application (Section 5) that relates data from distinct data sets by only manipulating control points.

To the best of our knowledge, orthogonal mapping theory has never been used in the context of multidimensional projection. Furthermore, this is the first time that multidimensional projection is explicitly employed to interactively correlate instances from data sets that have no connection.

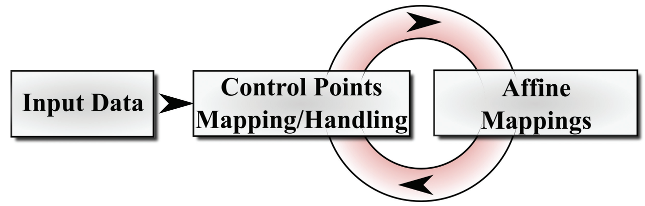


Fig. 2. The three main modules that compose the LAMP's framework.

2 Related Work

Multidimensional projection methods aim at mapping instances from a high dimensional space to the visual space so as to preserve distances as much as possible. Most MP methods derive from the multidimensional scaling (MDS) theory, a family of techniques that consider only distance information between pairs of instances to perform dimensionality reduction, thus making Cartesian coordinates for the original data unnecessary.

In order to better contextualize our approach and highlight its particularities we organize the existing MP methods into two main groups, global and local.

Global methods map data from a high-dimensional space to the visual space using a single transformation. Good examples of global methods are techniques based on spectral decomposition, which compute the embedding coordinates for each data instance from eigenvectors of a transformation applied to a dissimilarity matrix (symmetric matrix containing the distance between each pair of instances) [34]. Aiming at reducing the high computation costs (typically $O(n^3)$, where n is the number of instances to be projected) associated with the eigendecomposition Roweis and Saul [28] proposed an $O(n^2)$ algorithm that combines local fitting and global linear mapping. Sample instances were exploited by Brandes and Pich [3] and de Silva and Tenenbaum [11], achieving an $O(k^3+kn)$ algorithm, where k is the number of samples used to build the mapping. Subset of samples has been further exploited by Faloutsos and Lin [12] so as to obtain an $O(n)$ dimensionality reduction scheme. Tenenbaum et al. [33] proposed an $O(n)$ algorithm using a geometric framework. Multiscale matrix representation was employed by Belkin and Niyogii [1] and Koren et al. [20] towards reducing the computational burden, however, the lack of flexibility to enable user interaction hamper the effective use of spectral decomposition methods in visualization-oriented interactive applications.

First proposed by Kruskal [21], nonlinear-optimization-based techniques comprise the class of global methods that accomplish the mapping to visual space by finding a minimum for an energy function, usually called *stress function*. Optimization methods tend to be computationally expensive ($O(n^2)$), even when using efficient numerical solvers [5]. In order to reduce computational costs, Pekalska et al. [27] proposed a technique that first embeds a subset of samples in the visual space by optimizing a stress function and then places the remaining instances using a global linear mapping, resulting in an $O(k^3+kn)$ algorithm. Although more efficient than other optimization-based methods, Pekalska's approach is not flexible enough to support interactive applications while still requiring a minimum number of sample points equivalent to the dimension of the data.

Least Squares Projection [25] (LSP) is a two-step global technique that also uses a non-linear scheme to first position a subset of the samples in the visual space, mapping the remaining instances through a Laplacian-like operator, resulting an $O(k^2+n^2)$ algorithm. LSP makes use of a global neighborhood graph from which a large sparse linear system is derived. LSP allows for modifying the projection by manipulating the position of samples in the visual space, however, LSP's global nature limits the amount a projection can be changed. The same limitation can be observed in the recent linear mapping called PLMP [26] (whose complexity is linear), which also makes use of a subset of samples to define a global linear map. Likewise Pekalska's approach, PLMP also requires a minimum number of samples in order to accomplish the projection, disrupting interactivity.

In contrast to global approaches, the proposed LAMP technique can be set to behave as a local MP method, thus avoiding most of the problems inherent to global techniques.

Local methods make use of two main ingredients to perform the multidimensional projection, namely, the neighborhood information of each data instance and the location of a subset of samples a priori positioned in the visual space. More specifically, the mapping of each instance depends only on the sample points in its neighborhood, characterizing the local nature of the process.

The approach proposed by Chalmers [6] and its hybrid variants [19, 22, 32] first map the subset of samples to the visual space through a force-based scheme inspired in an analogy between stress function minimization and mass-spring systems. The neighborhood structure of each instance is then leveraged to embed the remaining data in the visual space, resulting in an $O(cn)$ technique, where c is the number of iterations performed by the algorithm. Despite the effort to mitigate computational effort [15, 18], this family of methods is still prohibitive for interactive applications that deals with large data sets.

The recently published PLP method [23] uses a force-based scheme to place the subset of samples in the visual space. The remaining data instances are projected using several local Laplacian-like operators, which are built from disjoint local neighborhood graphs. Flexibility in terms of user interaction is the main quality of PLP, since the user can move sample points around so as to change the projection layout and group similar instances. Drastic changes are possible because the underlying local neighborhood graphs are rebuilt during user interaction. The continuous update of the local graphs, however, increases the computational cost and tends to produce rank-deficient local Laplacian systems, thus impacting in robustness.

The proposed LAMP technique holds a striking combination of properties that makes it unique among the local multidimensional projection methods. Besides being cost effective and highly precise, LAMP does not rely on neighborhood graphs and its mathematical formulation admits a quite reduced subset of sample instances as input, properties that favor robustness when dealing with highly interactive applications that involve large data sets.

3 The LAMP Method

Similar to local methods, LAMP makes use of a subset of samples, from now on called *control points*, and their location in the visual space. The information get from control points is used to build a family of orthogonal affine mappings, one for each instance to be projected. The user can manipulate the control points in the visual space so as to better organize them. Since the affine mappings follow the layout of the control points, the user can interactively steer the projection, as illustrated in Figure 2.

Subsection 3.2 details the strategies we shall adopt to select control points and provides an analysis of stability/robustness when the number of control points changes. The underlying mathematical tools used to compute each affine map is described below.

3.1 Computing the Affine Mappings

To fix notation, let x be an instance in a data set $\mathscr{X}$, $x \in \mathbb{R}^m$ and x_i be the i^{th} element of the subset of control points $\mathscr{X}_S = \{x_1, x_2, \ldots, x_k\}$ selected from $\mathscr{X}$. The counterpart of $\mathscr{X}_S$ in the visual space ($\mathbb{R}^2$ in our context) is denoted by $\mathscr{Y}_S = \{y_1, y_2, \ldots, y_k\}$.

Given an instance x, the Local Affine Multidimensional Projection technique maps x to the visual space by finding the best affine transformation $f_x(p) = pM + t$ that minimizes

$$\sum_i \alpha_i \|f_x(x_i) - y_i\|^2, \quad \text{subject to } M^T M = I \tag{1}$$

where matrix M and vector t are the unknowns, I is the identity matrix, and α_i are scalar weights defined as:

$$\alpha_i = \frac{1}{\|x_i - x\|^2} \tag{2}$$

The minimization problem (1) is similar to that employed in "as-rigid-as-possible" image deformation [30]. However, in contrast to image deformation applications, where the affine transformations are from $\mathbb{R}^2$ to $\mathbb{R}^2$, we look for affine maps taking points from $\mathbb{R}^m$ to $\mathbb{R}^2$. Therefore, the explicit formulas used in image deformation do not apply in our context, which requires a more general minimization scheme.

Before deriving the mathematical tools used to solve (1), lets discuss some relevant aspects involved in such a minimization problem. The restriction $M^T M = I$ ensures that the resulting affine transformation behaves like a rigid transformation, that is, data can only be rotated and translated during the mapping process, avoiding scaling and shear effects. This behavior goes right back to what we need, namely, to preserve distances as much as possible during the multidimensional projection. If no constraint is enforced, the minimizer for the summation in (1) can be found through a conventional least square fitting procedure. However, in this case, errors in the positioning of control points are propagated by the local affine mappings, resulting in poorer quality projections. The orthogonality constraint ensures that errors introduced during the positioning of control points are not drastically propagated during the projection step. In other words, distortions are inevitable, the orthogonality constraint just keeps them as small as possible. We refer the interested reader to the book by Gower and Dijksterhuis [17], which provides a detailed discussion on how orthogonality and other more general constraints affect the outcome of minimization problems as the one stated in (1).

Another aspect to be observed is that the weights α_i depend on the point of evaluation, therefore, a distinct affine transformation is obtained for each instance x. Finally, in contrast to "as-rigid-as-possible" image deformation applications, we do not need to ensure continuity for the overall transformation, on the contrary, discontinuities may be highly desirable to better keep apart uncorrelated data instances during projection. This flexibility allows us to restrict the summation in (1) to take into account only control points in a neighborhood of x, rendering the process truly local. In fact, the larger the number of samples considered in the summation the less local is the projection of x. This fact will be exploited in our formulation and better discussed in Sections 3.2 and 4.

By taking partial derivatives with respect to t equal to zero, one can write t in terms of M as

$$t = \tilde{y} - \tilde{x}M, \quad \tilde{x} = \frac{\sum_i \alpha_i x_i}{\alpha}, \quad \tilde{y} = \frac{\sum_i \alpha_i y_i}{\alpha} \tag{3}$$

where $\alpha = \sum_i \alpha_i$. Therefore, the minimization problem (1) can be rewritten as

$$\sum_i \alpha_i \|\hat{x}_i M - \hat{y}_i\|^2, \quad \text{subject to } M^T M = I \tag{4}$$

where $\hat{x}_i = x_i - \tilde{x}$ and $\hat{y}_i = y_i - \tilde{y}$.

The minimization problem (4) can be expressed in matricial form

$$\|AM - B\|_F, \quad \text{subject to } M^T M = I \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and matrices A and B are given by

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{x}_1 \\ \sqrt{\alpha_2}\hat{x}_2 \\ \vdots \\ \sqrt{\alpha_k}\hat{x}_k \end{bmatrix}, \quad B = \begin{bmatrix} \sqrt{\alpha_1}\hat{y}_1 \\ \sqrt{\alpha_2}\hat{y}_2 \\ \vdots \\ \sqrt{\alpha_k}\hat{y}_k \end{bmatrix} \tag{6}$$

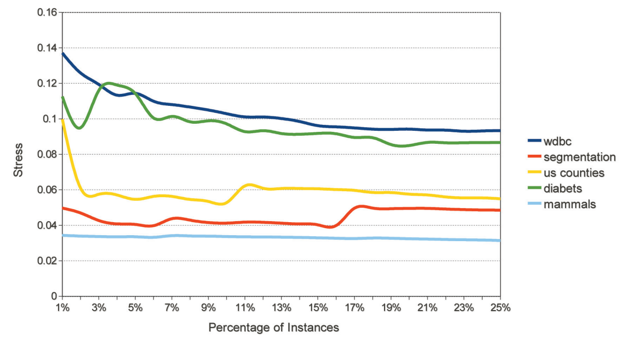


Fig. 3. The stress produced by LAMP when the number of control points ranges from 1% to 25% of the total of instances in the data set (see Table 1 for details about the data sets).

The minimization problem (5) is a typical example of the so called *Orthogonal Procrustes Problem* [17], whose solution is known to be

$$M = UV, \quad A^T B = UDV \tag{7}$$

where UDV is the singular value decomposition (SVD) of $A^T B$. Once M has been computed, the projection y of x is accomplished by

$$y = f_x(x) = (x - \tilde{x})M + \tilde{y} \tag{8}$$

At first glance, one may think that the calculation of a SVD decomposition for each data instance is too costly to be employed in an interactive application. However, $A^T B$ is indeed a $m \times 2$ matrix (only two columns), so it can be decomposed very quickly with compact SVD packages [2] ($O(k)$ operations), resulting in an algorithm with computational complexity equal to $O(kn)$. As we show in Section 4, besides resulting in highly accurate mappings, the mathematical construction described above turns out to be also competitive in terms of computational times.

Algorithm 1 (the LAMP algorithm) summarizes the main steps involved in the computation of the affine transformations used to map instances from a high-dimensional space to the visual space.

Algorithm 1 The LAMP's algorithm.

Require: Data set $\mathscr{X}$, control points $\mathscr{X}_S$, and the mapping $\mathscr{Y}_S$ of $\mathscr{X}_S$.
for each $x \in \mathscr{X}$ **do**
compute weights α_i // Equation (2)
compute $\tilde{x}$ and $\tilde{y}$ // Equation (3)
build matrices A and B // Equation (6)
compute the SVD decomposition UDV from $A^T B$
make $M = UV$
compute the mapping $y = (x - \tilde{x})M + \tilde{y}$
end for

3.2 Control Points Analysis

There are two main aspects to be observed when dealing with the control points $\mathscr{X}_S$. The first aspect relates to the number of control points that make up $\mathscr{X}_S$. Techniques such as PLMP [26], Pekalska's [27], and PLP [23] have limitations on the minimum number of control points to be employed. PLMP and Pekalska's method, for example, requires a number of control points (at least) equal to the dimension of the data, while PLP demands a minimum number of control points in each local neighborhood graph. In practice, those methods make use of $k = \sqrt{n}$ control points to build the mappings, where n is the total number of instances in the data set.

The LAMP technique, however, is very robust with respect to the number of control points, presenting low distortion even when a reduced number of control points is used, as shown in Figure 3. Notice that the stress function, given by $\frac{\sum_{ij}(d_{ij} - \bar{d}_{ij})^2}{\sum_{ij} d_{ij}^2}$ (d and $\bar{d}$ are the distance

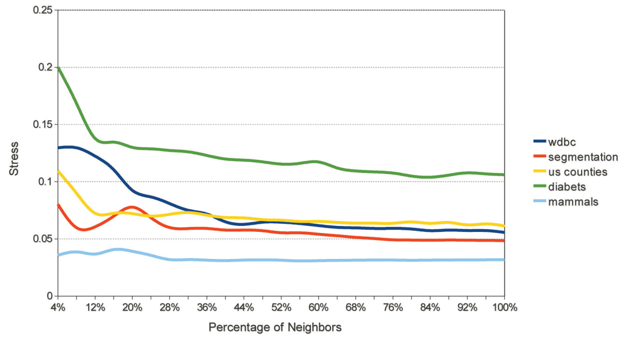


Fig. 4. Stress vs. percentage of nearest neighbors used to build the affine mappings. 100% means that all control points are used to build the mappings.

Table 1. Data sets used in the comparisons, from left to right the columns correspond to the data set name, size, dimension (number of attributes), and source.
⋆ http://www.cs.umd.edu/hcil/hce/examples/application_examples.html

Name	Size	Dim	Source
Wdbc	569	30	[14]
Diabetes	768	8	[14]
Segmentation	2,100	19	[14]
US counties	3,028	14	⋆
Isolet	6,238	617	[14]
Letter rcn	20,000	16	[14]
Mammals	50,000	72	[14]
Viscontest	200,000	10	[35]

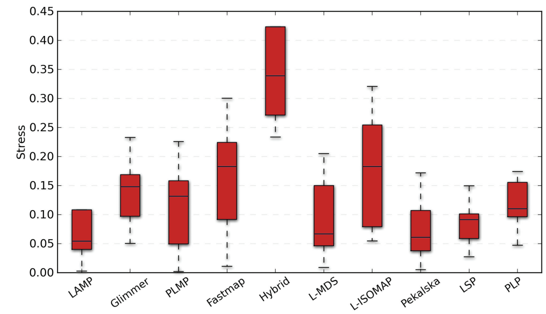
between instances p_i and p_j in the original and visual space), does not decay considerably when the number of control points increases, showing that LAMP can robustly map instances to the visual space using only a few control points. We are using an accurate force-based scheme [32] to place randomly selected control points in the visual space. The use of the force-based scheme is computationally viable because only a reduced number of control points need to be placed in the visual space, not significantly impacting in the overall performance of LAMP.

Besides the number of control points in $\mathscr{X}_S$, the number of terms in the summation in (4) can be tuned to modify the mapping behavior. As mentioned in the previous section, the overall mapping may become discontinuous when the summation does not go through all control points in $\mathscr{X}_S$. However, discontinuities may help to preserve groups (see Section 4) as well as to improve the quality of the projection. Figure 4 supports this assertion, showing that when the number of control points in the neighborhood of each instance x increases the stress energy still remains at low levels.

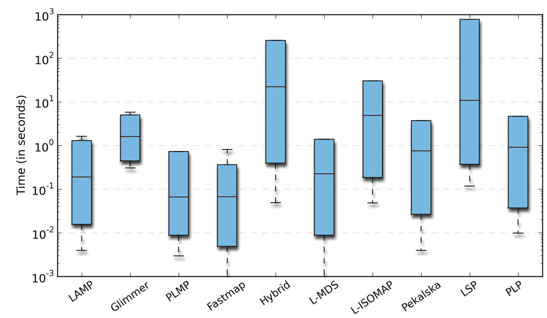
4 Results and Comparisons

All the results presented in this section were produced by an Intel® Core™ i7 CPU 920 2.66GHz, with an NVIDIA® Quadro FX 3800 video card and 8GB of RAM memory. LAMP is implemented in Java, and the JBlas numerical library [4] is used to compute the compact SVD decomposition.

In order to confirm the quality of the proposed technique we provide two different sets of comparisons. The first set aims at assessing LAMP's performance with respect to accuracy and computing time. We compare LAMP against 9 existing techniques employing eight data sets with comprehensive variation of size and data dimensionality (see Table 1). The techniques employed in the comparisons were chosen based on two criteria, they either present good performance in terms of stress and/or time or they use a subset of samples to carry out the mapping. The former criterion allows to compare LAMP against the most efficient MP methods while the second criterion confronts LAMP with methods that permit user intervention.



(a)



(b)

Fig. 5. Stress and computational times boxplots.

More specifically, Glimmer [18] has been chosen due to its good performance in terms of stress while Fastmap [12] is known to be a fast projection method. PLP [23], PLMP [26], Hybrid [19], L-MDS [11], L-Isomap [10], Pekalska [27], and LSP [25] are methods that present good stress/time results and also rely on a subset of samples to accomplish the multidimensional projection.

Boxplots in Figure 5(a) show that LAMP is one of the most accurate techniques, being comparable to highly precise methods such as Pekalska. The *original-distance* $\times$ *projected-distance* scatter plots presented in Figure 6 provide a visual tool to assess LAMP's accuracy. Notice that LAMP gives rise to almost 45^o diagonal layout in almost all test cases, implying that the original distances are well preserved in the visual space. The same is not true for other projection methods such as Hybrid and L-Isomap, which result in a spread distribution of points. Figure 5(b) also shows that LAMP is quite competitive in terms of computational times, being comparable to state-of-art methods such as PLP. In fact, LAMP is only slower than PLMP and Fastmap, techniques well known for their low computational cost.

The comparisons presented above confirm the accuracy and computational efficiency of LAMP. However, LAMP has been conceived to be interactive, that is, it should allow for the user to dynamically interfere in the projection. In order to analyze the LAMP's effectiveness in producing mappings that follow the control points layout provided by the user, we devise a second round of comparisons. Two distinct quantitative measures have been used to evaluate the mappings after interactive control points manipulation, namely, *neighborhood preservation* and *silhouette coefficient*. Given an instance x, the former measure gauges the percentage of the k-nearest neighbors of x that still remain neighbors in the visual space. The silhouette coefficient, which was originally proposed for evaluating clustering algorithms [31], measures both the cohesion and separation between grouped instances. The cohesion a_x of x is calculated as the average of the distances between x and all other instances belonging to the same group as x. The separation b_x is the minimum distance between x and all other instances belonging to other groups. The silhouette of a projection is given by $Silh = \frac{1}{n}\sum_{x\in\mathscr{X}} \frac{(b_x-a_x)}{max(a_x,b_x)}$ where n is the number of instances. Notice that $Silh$ ranges in the interval $[-1,1]$ and the larger the value of $Silh$ the better is the cohesion and separation.

The image in Figure 7(a) shows the force-based mapping of control points picked out from the Segmentation data set. The result of the

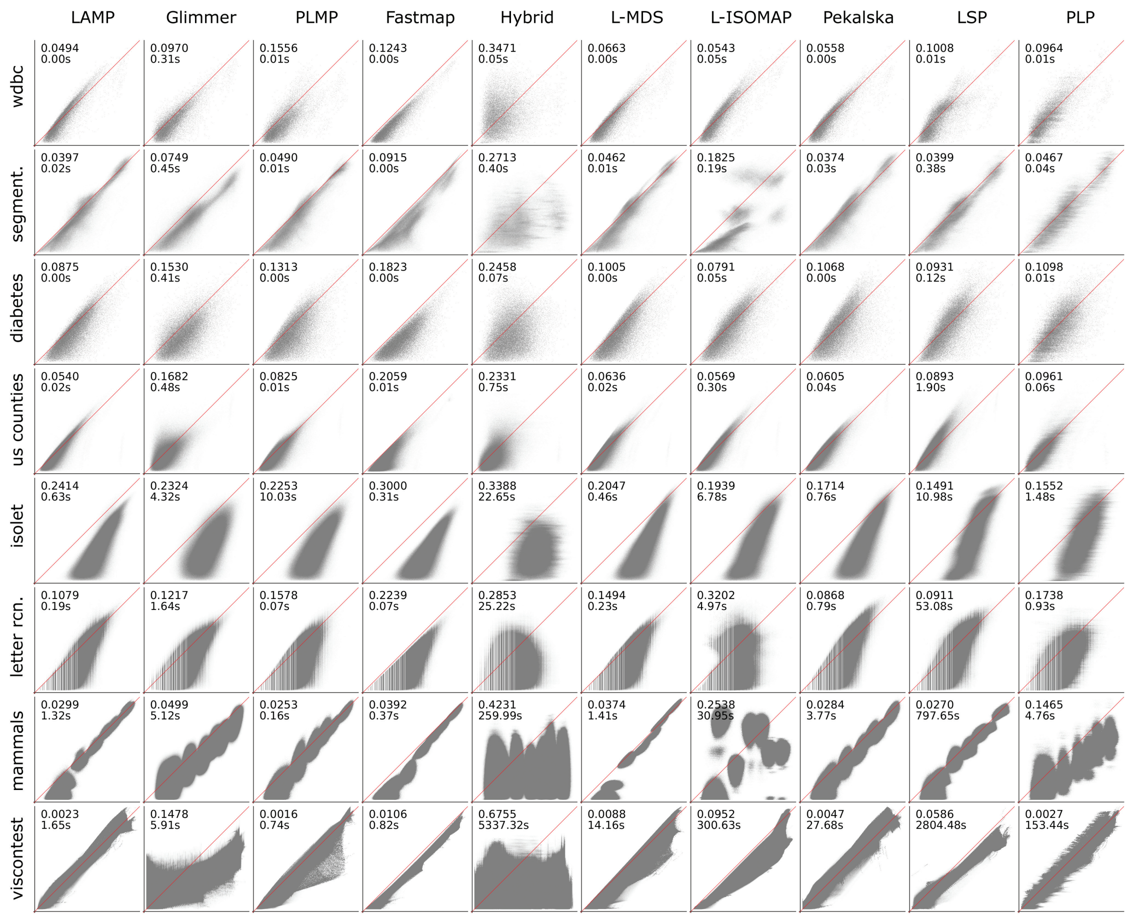
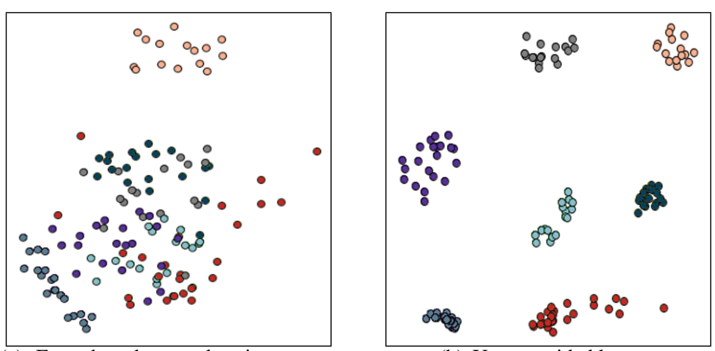
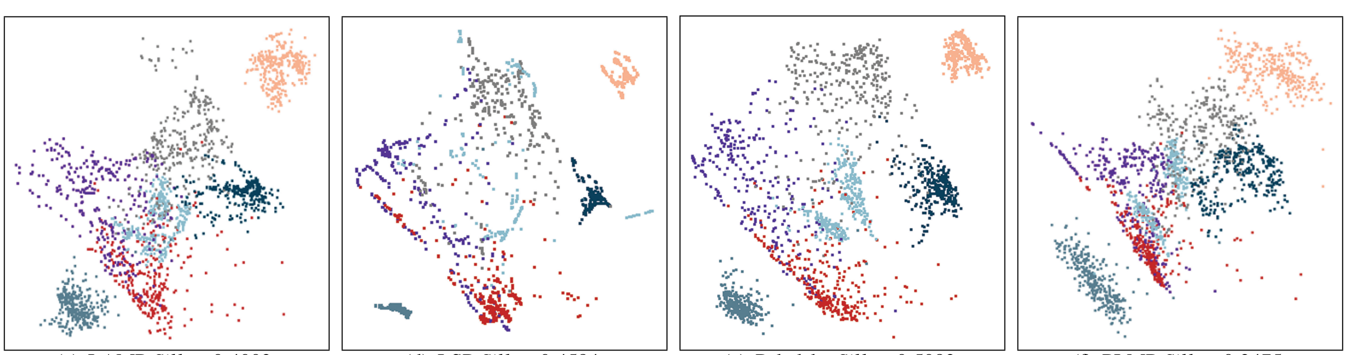


Fig. 6. *original-distance* × *projected-distance* scatter plots. Top-left numbers correspond to normalized stress and computational time (seconds).



(a) Force-based control points mapping

(b) User provided layout



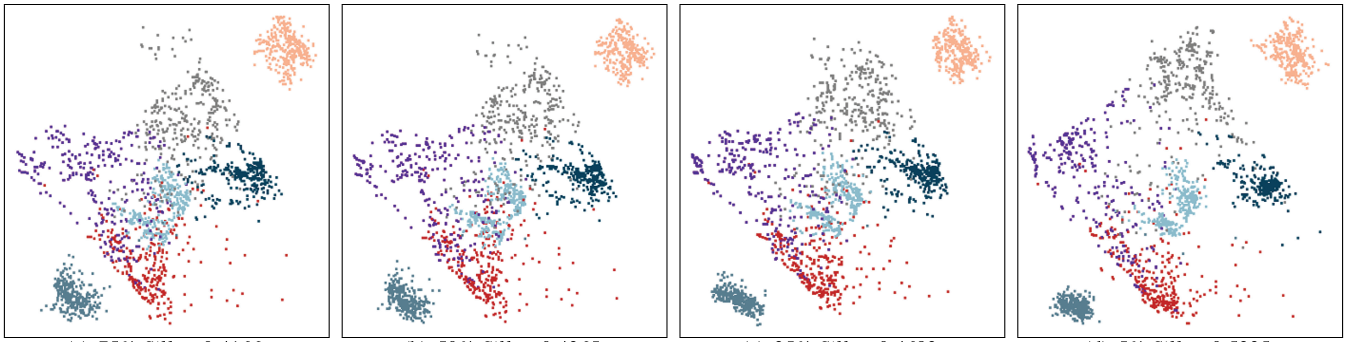
(c) LAMP $Silh = 0.4003$

(d) LSP $Silh = 0.4584$

(e) Pekalska $Silh = 0.5083$

(f) PLMP $Silh = 0.2475$

Fig. 7. Projections produced by LAMP, LSP, Pekalska and PLMP from user handled control points.



(a) 75% $Silh = 0.4166$ (b) 50% $Silh = 0.4365$ (c) 25% $Silh = 0.4682$ (d) 5% $Silh = 0.5235$

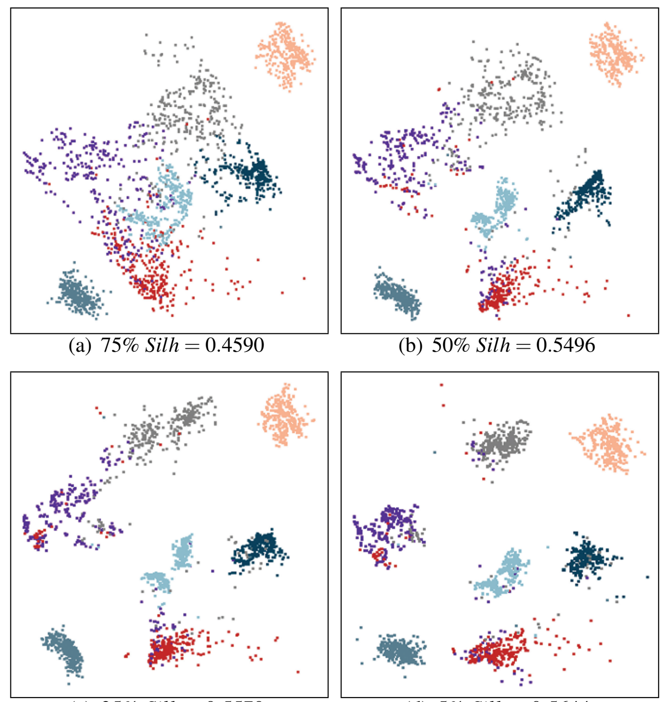
Fig. 8. Projections produced by LAMP varying the percentage of nearest control points used to build the mappings.

user intervention towards better grouping control points in the visual space is shown in Figure 7(b). Figures 7(c) to 7(f) depict the mapping produced by LAMP, LSP, Pekalska and PLMP respectively, all of them using the control points configuration shown in Figure 7(b). LAMP has used all control points to compute the affine map f_x for each instance x. As one can see, the *Silh* coefficient of LAMP is not as good as the ones produced by LSP and Pekalska, meaning the groups are better preserved by those two methods. However, the situation changes when LAMP makes use of the nearest control points of x to build the mapping f_x, as shown in Figures 8(a) to 8(d). Notice that the *Silh* coefficient increases consistently when 75%, 50%, 25%, and 10% of the nearest control points are used by LAMP to build the mappings, reaching a silhouette value higher than the three other global methods.

Neighborhoods can also be defined from information computed in the the visual space. The PLP technique, for example, relies on distances computed in the visual space to build neighborhood graphs from which the projection maps are derived. As shown in [23], the use of distances in the visual space leverages the capability of steering the projection according to the control point positions. This fact encouraged us to adapt LAMP so as to take into account 2D information when building the affine mappings. Given an instance x, let x_i be the control point nearest to x in $\mathbb{R}^m$. Rather than use control points in the neighborhood of x, LAMP can be modified to take into account the control points x_j whose images y_j are closer to the image y_i of x_i in the visual space. The mappings computed taking 2D neighborhood information into account push the projection of x toward the control points used in the computation of $f_x(x)$, which may not necessarily be neighbors of x in the original space.

The use of 2D information renders LAMP highly sensitive to the position of control points in the visual space, producing mappings that follow the layout of the control points very closely. This fact can be observed in Figures 9(a) to 9(d), where mapped instance becomes progressively more grouped when the percentage of nearest neighbors considered in the construction of the f_x ranges from 75% to 5% and neighbors are defined based on the visual space. The silhouette coefficient confirms that LAMP preserves groups nicely when 2D information is considered, outperforming PLP (see Figure 11) considerably. The superiority of LAMP can also be observed in the neighborhood preservation graphs presented in Figure 10. In fact, LAMP was able to preserve, on average, more than fifty percent of neighbors during projection.

To close this section, we show the robustness of LAMP when facing a reduced number of control points. As depicted in Figure 12(a), three control points were randomly selected in each one of the seven classes that composes the data set, being interactively placed in the visual space so as to keep distinct classes clearly separated. Figure 12(b) shows the mapping produced by LAMP using such a reduced set of control points (neighborhoods were defined from the visual space). Notice that even using a few control points (21 in total) LAMP was able to project the data set consistently, giving rise to a silhouette coefficient comparable to the one produce by PLP with much more



(a) 75% $Silh = 0.4590$ (b) 50% $Silh = 0.5496$

(c) 25% $Silh = 0.5570$ (d) 5% $Silh = 0.5644$

Fig. 9. Projections produced by LAMP from neighborhoods computed in the visual space. From left to right, the result of using 75%, 50%, 25%, and 5% percent of the nearest control points computed from the visual space.

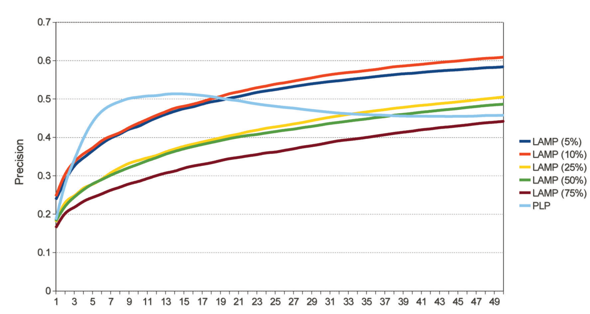


Fig. 10. Neighborhood preservation for PLP and LAMP.

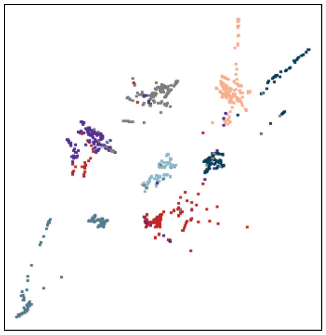
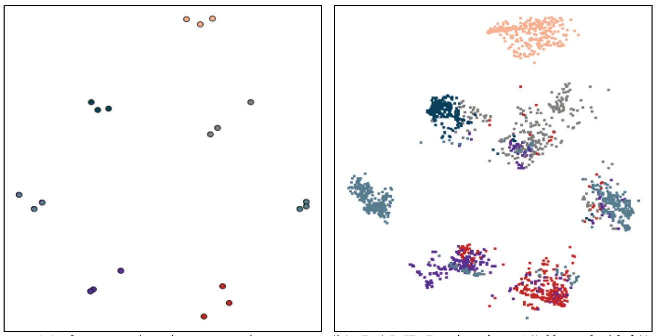


Fig. 11. Projection produced by PLP using distances in the visual space and the control points configuration shown in Figure 7(b) ($Silh = 0.4411$)



(a) 3 control points per class. (b) LAMP Projection ($Silh = 0.4361$)

Fig. 12. LAMP Projection (neighborhood in R^2) using only a few control points, 3 per class (21 in total against 137 needed to run the PLP).

points (Figure 11). We emphasize again that PLP, PLMP and Pekalska are not able to accomplish mappings using such a few control points. Moreover, due to their global nature, PLMP and Pekalska can not accomplish a drastic separation of instances such as the one produced by LAMP in Figure 9(d).

Robustness to deal with a reduced number of control points is a highly desirable property, mainly for interactive applications as the one we exploit in next section.

5 User-Assisted Data Correlation and Grouping

Incorporating user knowledge into the mapping by dynamically interacting with projected data is a useful functionality that can be exploited in many data visualization problems. In particular, we take advantage of the flexibility and robustness of LAMP in two applications, data correlation and document analysis.

User-assisted data correlation application aims at relating instances from data sets that do not have any connection. The idea is to start with a reduced set of control points selected from distinct data sets and interactively manipulate these control points in the visual space so as to bring closer instances that must be correlated. Once control points from distinct data sets have been correlated, that is, grouped together in the visual space, the remaining instances from each data set are projected using the LAMP technique. Since the mapping produced by LAMP follows the control points configuration, instances from the different data sets that are projected close to each other in the visual space are expected to be correlated. LAMP becomes specially useful in this kind of application because it allows for the user to select and manipulate a quite reduced number of control points, making the interaction an easier task.

The visual data correlation framework described above can be applied in many different scenarios. In particular, we have implemented a system that correlates images and music. The idea is to associate certain genres of music to specific kinds of pictures to automatically create slide shows with sound (Figure 1 presents an illustration of this framework applied to correlate music and images extract from videos). The user starts by picking out music from different genres, for example, a couple of classic music and a few rock-n-rolls. From a data set of images, the user selects a few pictures belonging to distinct classes, such as pictures of cars, houses, and aircrafts. Images and music from other classes can also be selected to represent instances that should not be correlated. The user then interacts with the selected music and images so as to bring close, for example, houses and classic music in a group and cars, aircrafts, and rock-n-roll in a second group. LAMP will map pictures of houses in the same neighborhood where the classic music will be project, the same being valid for the other group. Therefore, when pictures of houses are taken to form a slide show, classic music mapped in their neighborhood are automatically selected to compose the soundtrack. Figure 13 illustrates the prototype system we have developed to accomplish the task described above, that is, to build a correspondence between image and music. The accompanying video provides a better and clearer idea of the effectiveness of LAMP when integrated in the proposed application as well as the ease with which image and sound can be correlated using that system.

We have used a database with 3,857 music tracks and *JAudio Tool* [16] to extract low-level features from `mp3` files, such as beat points, statistical summaries, and so on, resulting in vectors with 78 dimensions. The *Caltech database* contains 3,100 pictures [13]. We employ the *bag-of-visual features (BoVF)* [36] approach to compute the image features, rendering 150 features.

The image and sound correspondence system described above is only a proof-of-concept of a new visualization-based paradigm for correlating data sets. In fact, very little has been done towards developing visual mechanisms to correlate distinct data sets and LAMP, with its good properties, brings out new perspectives to this application.

Document analysis is another application where LAMP can play an important part. Textual documents are typically embedded in very high dimensional spaces, which makes metrics such as Euclidean and cosine quite inefficient to discriminate documents. Therefore, the user skill is of paramount importance in order to organize and group documents as to their similarity. However, in order to identify similar documents the user has to ready the summary (set of key words) associated with each document, grouping the ones whose key words match closely. It is not difficult to realize that this is a tough task, even when handling a moderate number of documents. Therefore, techniques such as PLP and PLMP, which require a large number of control points to accomplish projections, are impracticable in this context. LAMP, however, can perform projections using a reduced number of control points, enabling the user to handle just a few documents to get good projection results, as shown in Figure 14. In the application depicted in Figure 14, a document collection composed by 675 scientific papers from four distinct areas, namely, Case-Based Reasoning (CBR), Inductive Logic Programming (ILP), Information Retrieval (IR), and Sonification (SON), has been handled by LAMP. Each document is embedded into a high dimensional space employing the vector space model approach [29], which extracts the frequency of relevant terms from the title, abstract, authors and references of each document to embed it in high dimensional space (390 dimensions in our case).

Figure 14(a) shows the result of placing, in the visual space, 12 control points picked out from the document data set (3 control points from each document class) using a force-based scheme [32]. Notice that documents from the same class are not properly grouped together, resulting in a tangled mapping (Figure 14(b)). Since few control points have to be handled, the user can easily identify the similar ones, arranging them next to each other in the visual space, as shown in Figure 14(c). The final projection, depicted in Figure 14(d), nicely preserves the user provided grouping, as attested by the silhouette coefficient.

6 Discussion and Limitations

The comparisons presented in Section 4 clearly show the effectiveness of the LAMP technique, surpassing, in requisites such as accuracy, robustness, and flexibility, state-of-art methods. The superior performance of LAMP is a consequence of the solid mathematical founda-

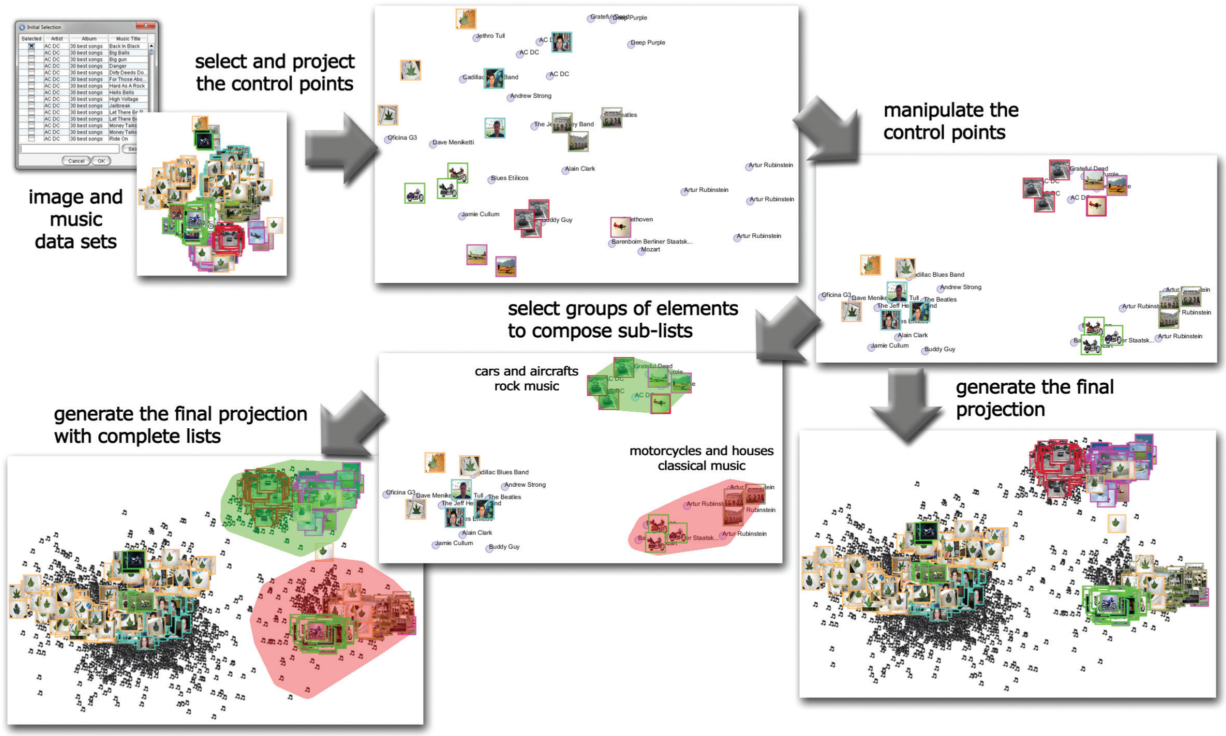
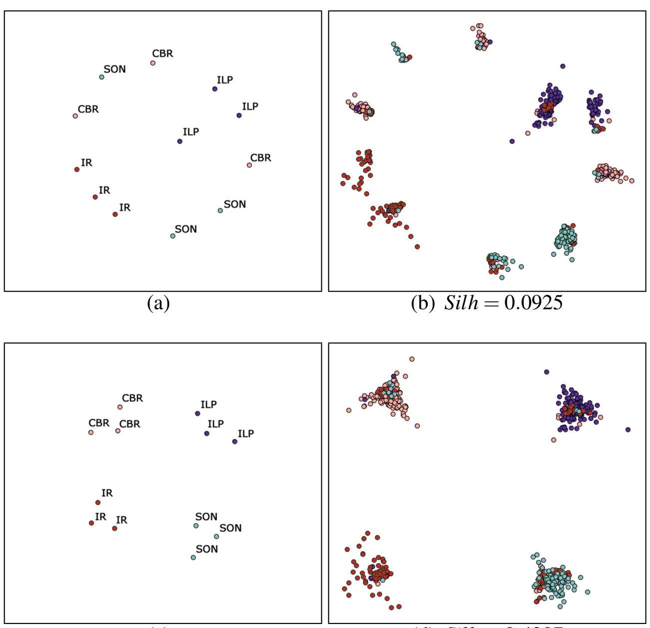


Fig. 13. Image and music correlation: A few representatives (control points) of music and pictures are selected from the corresponding data sets (top left) and placed in the visual space (top middle). User interacts with the sample pictures and music so as to bring close the ones to be correlated (top right). LAMP maps music and picture according to the user provided grouping (bottom right). The user can brush multiple regions in the visual space (bottom middle), each one corresponding to the pictures and music the will make up a slide show (bottom left).



(a) (b) $Silh = 0.0925$

(c) (d) $Silh = 0.4207$

Fig. 14. Due to the high dimensional nature of textual document data, the force-based scheme can not properly group similar instances in the visual space (a), resulting in a tangled mapping (b). Since LAMP supports few control points, the user can easily identify and group similar textual instances (c), resulting in a better projection (d). Colors are used to highlight documents belonging to the same class, but the class information is not used by the system.

tion it relies on, which ensures distance preserving and versatility towards incorporating user knowledge into the projection process. Simplicity is another strength of LAMP, which essentially requires a SVD decomposition library to be implemented.

The capability of accurately mapping instances using a few control points is another unique property of LAMP that many applications can benefit from. This property allied to the possibility of interactively changing the position of control points in the visual space render LAMP a very attractive method for applications such as visual exploration of documents as well as the novel visualization-based data correlation tool described in the previous section. In fact, the proposed framework for data correlation has enormous potential and can easily be adapted to work in social networks and scientific data.

In our experiments we notice that more "pleasant" layouts are produced when the control points x_i and their image y_i are in the same scale. Therefore, we normalize data in the original space as well as the control points position in the visual space. Although such a sensitivity to the difference in scale is not a serious limitation, it is worth to heed the scales when implementing LAMP. Choosing the ideal number of neighbors to produce the desired layout is another aspect we have to investigate more deeply. An alternative to the k-nearest neighbors scheme employed in our implementation would be to define a radius of influence to each control point. However, finding the appropriate radius to be assigned to each control point is not an easy task either, being an issue to be investigated.

7 Conclusion

In this work we proposed a novel projection technique called Local Affine Multidimensional Projection (LAMP), which is shown to be very effective for interactive applications. LAMP has a solid mathematical foundation which ensures robustness and versatility. The set of comparisons we provided shows that LAMP outperforms existing projection methods in terms of stress minimization while still being competitive regarding computational times. Moreover, the potential use of

LAMP to support visualization-based data correlation opens new possibilities for applications which could not be efficiently addressed until now. Therefore, flexibility, effectiveness, and ease of implementation render LAMP one of the most attractive multidimensional projection methods for handling high-dimensional data. We are currently investigating better mechanisms to assess the impact of user interaction in the quality of the projection, since stress is not a useful measure after interaction.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments. This research has been funded by Fapesp-Brazil, CNPq-NSF, and CNPq-Brazil.

REFERENCES

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

[2] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.*, 28:135–151, June 2002.

[3] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In M. Kaufmann and D. Wagner, editors, *LNCS*, volume 4372, pages 42–53. 2007.

[4] M. L. Braun, J. Schaback, M. L. Jugel, and N. Oury. jBlas: Linear algebra for java, 2011.

[5] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications*, 13:149–171, 2006.

[6] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *IEEE Visualization*, pages 127–ff., 1996.

[7] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus. *IEEE Trans. Vis. Comput. Graph.*, 15:1161–1168, 2009.

[8] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu. Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, pages 42–53, 2010.

[9] J. Daniels, E. W. Anderson, L. G. Nonato, and C. T. Silva. Interactive vector field feature identification. *IEEE Trans. Vis. Comput. Graph.*, 16:1560–1568, 2010.

[10] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, 2003.

[11] V. de Silva and J. B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford, 2004.

[12] C. Faloutsos and K. Lin. FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases. In *ACM SIGMOD*, pages 163–174, 1995.

[13] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, 2003.

[14] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[15] Y. Frishman and A. Tal. Multi-level graph layout on the GPU. *IEEE Trans. Vis. Comput. Graph.*, 13:1310–1319., 2007.

[16] I. Fuinaga and D. McEnnis. On-demand metadata extraction network (OMEN). In *ACM/IEEE-CS Joint Conf. on Digital Libraries*, pages 346–346, 2006.

[17] J. Gower and G. Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004.

[18] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Vis. Comp. Graph.*, 15(2):249–261, 2009.

[19] F. Jourdan and G. Melançon. Multiscale hybrid MDS. In *Information Visualisation*, pages 388–393, 2004.

[20] Y. Koren, L. Carmel, and D. Harel. ACE: A fast multiscale eigenvectors computation for drawing huge graphs. In *IEEE Information Visualization*, pages 137–144, 2002.

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

[22] A. Morrison, G. Ross, and M. Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *IEEE Information Visualization*, pages 152–158, 2002.

[23] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum (EuroVis'2011)*, 30(3):1091–1100, 2011.

[24] F. V. Paulovich, L. G. Nonato, and R. Minghim. Visual mapping of text collections through a fast high precision projection technique. In *International Conference on Information Visualization*, pages 282–290, 2006.

[25] 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 Trans. Visual. Comp. Graph.*, 14(3):564–575, 2008.

[26] F. V. Paulovich, C. T. Silva, and L. G. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Trans. on Vis. Comp. Graph.*, 16(6):1281–1290, 2010.

[27] E. Pekalska, D. de Ridder, R. P. W. Duin, and M. A. Kraaijveld. A new method of generalizing Sammon mapping with application to algorithm speed-up. In M. Boasson, J. A. Kaandorp, J. F. M. Tonino, and M. G. Vosselman, editors, *Annual Conf. Advanced School for Comput. Imag.*, pages 221–228, 1999.

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

[29] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[30] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, 2006.

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

[32] E. Tejada, R. Minghim, and L. G. Nonato. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization*, 2(4):218–231, 2003.

[33] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[34] W. S. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1965.

[35] 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.

[36] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *International Workshop on Multimedia Information Retrieval*, pages 197–206, 2007.