iLAMP: Exploring High-Dimensional Spacing through Backward Multidimensional Projection

Elisa Portes dos Santos Amorim[1*] Emilio Vital Brazil[1*] Joel Daniels II[2†] Paulo Joia[3‡]
Luis Gustavo Nonato[3‡] Mario Costa Sousa[1*]

[1]University of Calgary, [2]NYU Polytechnic Institute, [3]University of Sao Paulo

ABSTRACT

Ever improving computing power and technological advances are greatly augmenting data collection and scientific observation. This has directly contributed to increased data complexity and dimensionality, motivating research of exploration techniques for multidimensional data. Consequently, a recent influx of work dedicated to techniques and tools that aid in understanding multidimensional datasets can be observed in many research fields, including biology, engineering, physics and scientific computing. While the effectiveness of existing techniques to analyze the structure and relationships of multidimensional data varies greatly, few techniques provide flexible mechanisms to simultaneously visualize and actively explore high-dimensional spaces. In this paper, we present an *inverse linear affine multidimensional projection*, coined iLAMP, that enables a novel interactive exploration technique for multidimensional data. iLAMP operates in reverse to traditional projection methods by mapping low-dimensional information into a high-dimensional space. This allows users to extrapolate instances of a multidimensional dataset while exploring a projection of the data to the planar domain. We present experimental results that validate iLAMP, measuring the quality and coherence of the extrapolated data; as well as demonstrate the utility of iLAMP to hypothesize the unexplored regions of a high-dimensional space.

1 INTRODUCTION

The visualization of high-dimensional datasets has been a recurrent research topic in the visualization community. New techniques that provide visual representations of such complex data are in constant development. High-dimensional datasets are, undoubtedly, of extreme interest as they appear in most scientific domains, such as engineering, physics and scientific computing. Most problems of these and other fields can be seen as a system whose output depends on a set of parameters which can be concatenated so as to form high-dimensional instances. Well-established high-dimensional visualization techniques have proven to be efficient tools to allow the comprehension of the relationship among various instances and/or parameters of the dataset. However, for most scientific applications, it is required not only to analyze existing data, but to create and evaluate new parameter combinations. Consider, for instance, the problem of encountering input parameter combinations that result in a particular system output. This is a very common problem in science and is usually modeled as optimization problems. Often, the parameter space that defines the regions of feasible solutions are very large and difficult to navigate. Automatic optimization methods are usually employed but fail on incorporating the user expertise and intuition to the process. Moreover, the problem's solution is often non-unique, i.e., several different parameter combinations suit the required restrictions, what makes an effective exploration of the parameter space a very important issue.

The technique proposed in this paper brings a new perspective for parameter space analysis and exploration by complementing a traditional and well established high-dimensional visualization technique with new interactive resources that allows for navigating and resampling specific regions of the space. More specifically, we empower the multidimensional projection technique called LAMP [25] with the new interactive functionalities, rendering it a more versatile visual analysis tool. The proposed technique converts the multidimensional projection method into a really interactive tool which allows the user to inspect the visual space and use it to create new high dimension instances.

But why to plug the new interactive mechanism into a multidimensional projection method? It is not trivial to incorporate extrapolation mechanisms into conventional high-dimensional data visualization techniques such as parallel coordinates and scatter plots [14]. The difficulty arises mainly because such techniques do not provide a mechanism to visualize the neighborhood structure of data under analysis, making harder the process of identifying similar instances. Sophisticated visualization techniques able to provide some neighborhood information rely on non-intuitive visual metaphors that hamper the user experience when exploring multidimensional spaces.

In contrast, the intrinsic properties of multidimensional projection (MP) techniques make the visualization and manipulation of neighborhood structures straightforward, motivating us to use MP as basis for our resampling mechanism. In fact, the method presented in this work takes advantage of the inherent properties of MP techniques, tailoring the novel interactive tool so as to visualize regions of interest while still enabling mechanisms to synthesize new data in those regions. Called *iLAMP (inverse-LAMP)*, our method allows for resampling the high-dimensional space through an interactive interface.

The resampling mechanism maps new points created by the user in the visual space back to the high-dimensional space while providing a broad view of the neighborhood where the new data is being created. iLAMP performs the backward projection through local affine mappings that preserve distances between the new samples as much as possible, as it follows the same concept presented on LAMP. Using the proposed scheme the user can interactively extrapolate instances in a dataset, generating synthetic multidimensional data out of existing projections.

We assess the robustness and accuracy of iLAMP by applying it in synthetic and manufactured data sets where errors can be measured and visualized. Moreover, the proposed methodology is employed in an optimization-oriented application whose goal is to figure out the location of local minima as well as to explore regions of the parameter space not visited by the optimization algorithm. In this case, iLAMP is used to generate new possible parameter combinations in unexplored regions of the original space. The new instances are used as starting point for further optimization.

*e-mail: {epdamori,evbrazil,smcosta}@ucalgary.ca
†e-mail:jdaniels@nyu.edu
‡email: {pjoia, gnonato}@icmc.usp.br

We can summarize the contributions presented in this paper as:

- *iLAMP*: A novel mechanism to map information from the visual space back to a high-dimensional space. The method builds upon the recently proposed MP technique called LAMP [25] to define affine mappings from the visual to the high-dimensional space (Section 3).
- *User-driven High Dimensional Space Exploration*: iLAMP allows the user to extrapolate existing data in an interactive manner, using visual feedback provided by the projection to generate new data that helps to further explore parameter spaces. We present an application scenario where iLAMP is used as an exploration tool for high-dimensional parameter spaces governing optimization problems (Section 5).

To the best of our knowledge, the methodology presented in this paper is the first one to enable a coherent connection between visual representation given by a multidimensional projection technique and interactive exploration/sampling of the high-dimensional data space.

2 Related Work

In this section we present an overview of traditional techniques that proposes the visualization of high-dimensional data. We classify multidimensional visualization techniques in three main categories: Non-Projective Visualization, Simple Projection Visualization, and Dimensionality Reduction Visualization.

2.1 Non-Projective Mappings

Multivariate visualization techniques, i.e. mapping high-dimensional data to a 1- or 2-dimensional visual space, have long been a focus of information visualization research. For instance, *parallel coordinates* is a popular technique to visualize and interactively explore multivariate data [22]. Each dimension of the data is represented as a parallel coordinate, describing a non-projective mapping of the *N*-dimensional space to the plane. A point in the N-dimensional dataset is represented as a polyline in parallel coordinate, connecting its value for each data dimension. *Star Coordinates* is a variation of parallel coordinates, in which the axis for each data dimension share a common origin [28, 29].

These techniques have been shown to be useful in cluster discovery and multi-factor analysis; however, visual clutter becomes problematic when exploring and visualizing large data. Hierarchical parallel coordinates perform clustering on the data, visualizing semi-transparent bands associated with each group [4, 16, 24]. While the clusters convey important information, outliers are lost within these visualizations, better exposed through focus+context visualizations [34]. While simple to construct and interact, the neighborhood relationships between pairs of points can be difficult to perceive within these non-projective techniques. In contrast, relative similarities between pairs of points are directly encoded in our visualizations.

2.2 Simple Projective Mappings

In contrast to parallel coordinates, *scatterplots* describe a simplistic projective view of the high-dimensional data [8]. A scatterplot is a standard plot, considering two variables of the high-dimensional data. Many visualization toolkits provide interactive scatterplot capabilities, including Tableau/Polaris [45] and GGobi [46]. Because scatterplots are limited in the number of dimensions they visualize in comparison to the size of most datasets, multiple plots are typically arranged into rows and columns forming *scatterplot matrices*.

Similar to parallel coordinates, it can become challenging to discern the important information while plotting large datasets. Statistical analyses and hierarchical group plotting [43, 44] highlight the relationships between and within different groups. Other research investigates automatically sorting the coordinates to determine salient correlations between different dimensions [40, 50]; as well as integrating these features with animation effects for scatterplot browsing and interaction [13]. Scatterplot matrices are simple to construct and interact; however, they require a profound knowledge of the multiple coordinated views. In contrast, similarities between different regions of the data are straightforward in our visualizations.

2.3 Dimensionality reduction mappings

In the context of this paper, *dimensionality reduction* solutions further extend scatterplots by encoding additional (if not all) dimensions of the original data within the 2D visualization. Often a dimensionality reduction solution will attempt to preserve distance relationships between pairs of points in the high-dimensional space through the mapping. When this is the case, the resulting scatterplots become extremely useful for visual analysis and exploration within the plane of similarities hidden in the high-dimensional data. Dimensionality reduction techniques can be either linear or non-linear mapping solution.

Linear projection routines determine a mapping of the data with a single transformation. Principal component analysis (PCA) [26] is a well-known example of dimensionality reduction that determines a number of orthogonal dimensions describing the maximal data variances. These dimensional vectors describe the dominant trends in the data. Other linear projection routines, including the linear squares projection (LSP) [38] and part-linear multidimensional projection methods [39], solve projections based on a non-linear mapping of a subset of the dataset. A recent LSP-based interactive system [20] avoids the computational complexity of the control point distribution by relying on user inputs. However, dense projections, due to few control points, causes the proposed system to be sensitive during the painting sessions.

Non-linear multi-dimensional scaling. Many dimensionality reduction techniques can be described as variants of multi-dimensional scaling (MDS) [9]. MDS techniques rely on relational measures between pairs of data samples and can ignore the original data coordinates. Sammon's mapping [42], a popular MDS technique, defines a function describing the error in distances between point pairs due to the projection, then iteratively reduces this residual. Modifications to the distance function, with threshold [11] and geodesic distances [48, 51], addresses the volatility of Sammon's mapping under large distances.

Spectral decomposition [49] is a global MDS routine, mapping high-dimensional data to a visual space by computing eigenvectors of the symmetric matrix encoding the distances between each pair of data samples. To address high computational costs associated with large datasets, extensions to spectral decomposition combine local fitting and global linear mapping [41], use landmark subsamples [5, 10], and develop multi-scale matrix representations [3, 30]. However, spectral decomposition lacks a flexibility that would enable user interactions, limiting application of such techniques in visual exploration frameworks.

One fundamental issue to improve the interactivity and visual insights over complex high-dimensional data is to layout its projection properly on the plane of the exploratory visual space. Force-based schema have been used to design graph layouts for visualization [12], with many variants [6, 27, 32, 33, 35, 37, 47]. These techniques operate similar to finding the equilibrium of a spring-mass system. Points are iteratively relaxed within the visual space to reduce distortions in relative distances between local neighbors as they are mapped from the high-dimensional space. Multivariate brushing of force-based graphs [23] enables user exploration of high-dimensional data; however, the expensive convergence required a static graph layout. While multilevel techniques [15, 21]

address this restriction, it remains a challenge for interactive support of very large datasets.

The mathematical foundation of the method described in this paper is based on the LAMP methodology [25]. LAMP is a multidimensional projection technique that derives from orthogonal mapping theory. It relies on local information to build affine transformations that map points from high to low dimension. iLAMP performs the inverse mapping, still using the same mathematical background.

Dimensionality reduction methods have been the foundation of several high-dimensional exploration and visualization tools. Feature exploration in multivariate scalar fields [23], vector fields [20], and text mining [7, 36] applications rely on the clustering nature of these projection techniques. While these techniques enable user interaction, they restrict the exploration tasks to analysis of the *existing* data. The same is true for techniques that propose the visualization of high-dimensional scalar functions, such as the works from Harvey and Wang [19], and Gerber et al [17]. These methods are valuable tools for optimization problems, but they do not allow the user sampling and extrapolation of the data. In contrast, the iLAMP method proposed in this paper, enables interactive visual exploration of the projected space as well as the backward projection to extrapolate data that could be, creating a robust and fast framework for visual exploration and analysis of high-dimensional data.

3 iLAMP

In this section we present the theory behind iLAMP. In short terms, iLAMP maps a point on the screen into a high-dimensional vector. The proposed method takes as input a high-dimensional dataset and its correspondent low-dimensional position, calculated by a multidimensional projection technique. With these datasets, iLAMP provides an interactive environment that allows the user to create new points in the visual space that contains the data projection. iLAMP computes an affine transformation that takes a point $\mathbf{p}$, defined by the user in the visual space, to a point $\mathbf{q}$ in the original high-dimensional space. The transformation matrix is constructed in such a way that the distances between the new point $\mathbf{q}$ and the high-dimensional instances are as close as possible to the distances between the user selected point $\mathbf{p}$ and the projected data.

Section 3.1 describes the mathematical details of the proposed technique, which is based on the recently proposed multidimensional technique LAMP [25]. Later, on Section 3.2, we discuss computational aspects of the method. Section 3.3 discusses how iLAMP handles *false neighborhoods* and *tears*, common artifacts of multidimensional projections.

3.1 Mathematical Formulation

Let $\mathbf{x}_i$ be an instance in dataset $X \subset \mathbb{R}^m$, and its correspondent instance $\mathbf{y}_i$ in dataset $Y \subset \mathbb{R}^2$; i.e.,$\mathbf{y}_i$ is the correspondent multidimensional projection of $\mathbf{x}_i$ in the visual space. Both sets, X and Y, are given as input to iLAMP, which uses this information to build local affine transformations $f : \mathbb{R}^2 \to \mathbb{R}^m$ to map a point $\mathbf{p}$ from the visual space to a point $\mathbf{q}$ in $\mathbb{R}^m$.

Given a point $\mathbf{p} \in \mathbb{R}^2$ and $k \in \mathbb{N}$, the first step in the iLAMP algorithm is to find the k closest neighbors to $\mathbf{p}$ among the instances in Y. All the subsequent calculations are done solely based on these k instances and their correspondent high-dimensional vectors encountered in dataset X. Let $Y_S = \{\mathbf{y}_1, \mathbf{y}_2 \ldots \mathbf{y}_k\}$ be the subset of Y that contains the k closest points to $\mathbf{p}$ and $X_S = \{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_k\}$ the dataset containing the correspondent high-dimensional instances on X.

iLAMP maps $\mathbf{p}$ from the visual space to the original high-dimensional space $\mathbb{R}^m$ by finding the affine transformation $f(\mathbf{p}) = \mathbf{p}M + \mathbf{t}$ that minimizes

$$\sum_{i=1}^{k} \alpha_i \| f(\mathbf{y}_i) - \mathbf{x}_i \|^2, \qquad \text{subject to } M^T M = I, \tag{1}$$

where matrix M and vector $\mathbf{t}$ are the unknowns, I is the identity matrix, and α_i are scalar weights defined as

$$\alpha_i = \frac{1}{\|\mathbf{y}_i - \mathbf{p}\|^2}. \tag{2}$$

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

$$t = \tilde{\mathbf{x}} - \tilde{\mathbf{y}}M, \qquad \tilde{\mathbf{x}} = \frac{\sum_{i=1}^{k} \alpha_i \mathbf{x}_i}{\alpha}, \qquad \tilde{\mathbf{y}} = \frac{\sum_{i=1}^{k} \alpha_i \mathbf{y}_i}{\alpha}, \tag{3}$$

where $\alpha = \sum_{i=1}^{k} \alpha_i$. The minimization problem described on (1) can be written as

$$\min_M \sum_{i=1}^{k} \alpha_i \| \hat{\mathbf{y}}_i M - \hat{\mathbf{x}}_i \|, \qquad \text{subject to } M^T M = I, \tag{4}$$

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

The minimization problem (4) can be written in matricial form as

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

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

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

The solution of (5) is given by

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

where UDV is the singular value decomposition (SVD) of $A^T B$. Once M has been computed, the mapping of a point $\mathbf{p}$ to the high-dimensional space is accomplished by

$$\mathbf{q} = f(\mathbf{p}) = (\mathbf{p} - \tilde{\mathbf{y}})M + \tilde{\mathbf{x}}. \tag{8}$$

The rationale behind the minimization problem (5) is to build affine mappings that respect the correspondence $\mathbf{y}_i \leftrightarrow \mathbf{x}_i$. Note that, if $\mathbf{p} \to \mathbf{y}_i$ the α_i weight assigned to $\mathbf{y}_i$, calculated on Equation (2), goes to infinity, i.e., the backward mapping of a projected data $\mathbf{y}_i$ is precisely its counterpart $\mathbf{x}_i$. Moreover, the constraint $M^T M = I$ enforces Euclidean distances (norms) to be preserved as much as possible during the mapping, as demonstrated bellow

$$\|M\mathbf{x}\|^2 = (M\mathbf{x})^T M\mathbf{x} = \mathbf{x}^T M^T M\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2.$$

Since the Euclidean norms are preserved, the orthogonal matrix M acts as an isometric transformation in the mapping. Such methodology results in a quite accurate mapping scheme, as shown in Section 4. Before attesting the quality of iLAMP, we discuss some computational and implementation aspects of the technique.

3.2 Computational Aspects and Implementation

In this section we discuss some implementation and computational aspects of the iLAMP technique. We provide a time analysis of the method, a discussion on the neighborhood of point **p** and interaction scheme of the proposed method.

Compact SVD and Time Analysis: The bottleneck of the iLAMP computation is the calculation of the SVD of matrix $A^T B$ (Equation (7)). Matrices A^T and B are $2 \times k$ and $m \times k$, respectively, what makes $A^T B$ a $2 \times m$ matrix. Since $A^T B$ contains only 2 rows, one may employ *compact* SVD methods to compute the required decomposition, reducing considerably the computational costs when compared to a full SVD scheme. In our implementation we use the LAPACK library [1] compact SVD routine.

The complexity of the iLAMP algorithm depends only on the number of neighbors k and in the dimensionality of the original data, as the calculation of the closest neighbors to point **p** can be accomplished using efficient methods, such as quadtrees. Figure 1 displays the data dimensionality vs. time in milliseconds spent by iLAMP to calculate 100 (one hundred) backward mappings, for 3 different values of k. The experiments were performed in a Intel Core i7-860 Processor (8M Cache, 2.80 GHz). Fifty runs of the algorithm were performed for each test case, and the illustrated results are the average time acquired. We observe that iLAMP is extremely fast, what proportionates an interactive real-time application. For instance, for $k = 100$ and $m = 450$, the average time spent to calculate 100 backward mappings was 263 milliseconds.

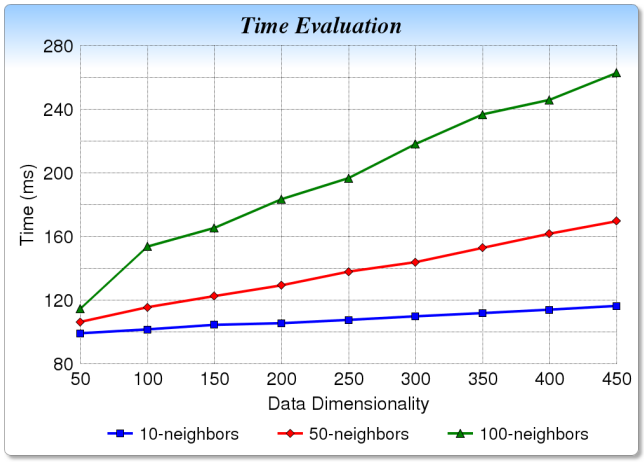


Figure 1: Time, in milliseconds, consumed in the generation of 100 samples using iLAMP.

Number of neighbors: The number of neighbors k is an important parameter for the iLAMP algorithm. It defines how many instances of the original dataset will be taken into account on the construction of the affine transformation matrix M that will map point $\mathbf{p} \in \mathbb{R}^2$ to a point $\mathbf{q} \in \mathbb{R}^m$. Thus, the value of k has a deep impact on the quality of the output transformation. In this work we chose k in a heuristic fashion. Although the ideal value of k varies from dataset to dataset, we experimentally inferred that good k values are usually found in the range of three and twenty.

Interaction scheme: The main component of our system is a screen that displays the multidimensional projection of a high-dimensional dataset. The displayed projection provides the visual feedback that allows the user to analyze the data and identify possible regions of interest to be further explored. For instance, these can be empty regions in the projection space that are close to instances of particular interest. Using direct point selection, via mouse or any pointing device, the user is able to extrapolate the data in such regions, by creating new points in the projection space and mapping them to the original space using the iLAMP technique. This allows the user to experiment what-if scenarios, exploring the parameter space in an interactive and intuitive way.

3.3 Handling false neighborhoods and tears

False neighborhoods and tears are artifacts that may appear in multidimensional projections. A false neighborhood occurs when a large distance in the original space is associated with a small distance in the projection space. This distortion falsely suggests a neighborhood of points that do not accurately reflect the original distribution of data. In contrast, a tear occurs when a small distance in the multidimensional space is associated with a large distance in the projection space. This distortion falsely conveys a large difference between nearby neighboring points.

Some projection techniques attempt to reduce these artifacts; however, in many cases they are unavoidable [2]. For instance, consider a dataset composed by samples of a hyper-sphere (or any other closed surface). The projection of such samples will be somehow distorted with the existence of false neighbors, tears, or both [31]. More generally, the presence of outliers in a dataset often lead to false neighborhoods in the projection.

If these artifacts are disregarded in the back-projection process, iLAMP-generated points may become distorted and end up in unexpected regions of the original space. So far, we have considered that the k closest points to **p** are searched among the instances of dataset Y. In fact, this is the most natural procedure, as **p** and $\mathbf{y} \in Y$ are defined in the same space $\mathbb{R}^2$. However, if false neighborhoods and/or tears are present in the projection, using only the low-dimensional information for this task can result in misleading back-projection mappings.

To accommodate for artifacts in the projection, it is possible to incorporate the multidimensional data in the neighborhood definition of a point **p**. The closest instance $y_i \in Y$ to **p** seeds the neighborhood search in the original space. The $k-1$ closest neighbors to the multidimensional instance $x_i \in X$ corresponding to y_i, $(x_{i1}, x_{i2}, x_{ik-1})$, define the neighborhood of **p**. This *high-dimensional neighborhood search* prevents the usage of a neighborhood set composed of false neighbors.

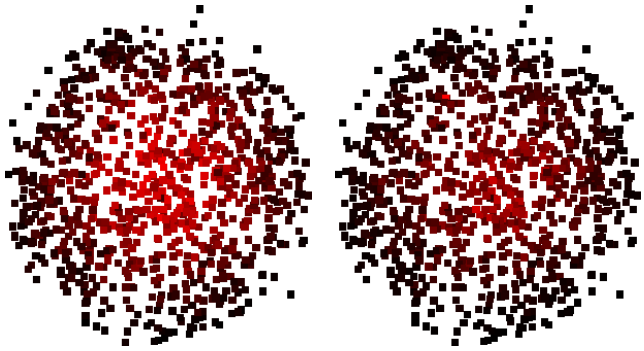
Note that we do not enforce the usage of the high-dimensional neighborhood for the back-projection mapping. Instead, we allow the user to decide whether to use low-dimensional information only, or to include the high-dimensional information. However, the user should have a way to assess the quality of the projection as a whole or of regions of interest in order to make an informed decision on how the neighborhood set will be formed.

We provide three visual tools to provide indications to the user of artifacts within the projection. These methods are based on proposed visualization techniques of false neighborhoods and tears [2]. In particular, Lespinats and Aupetit [31] propose two metrics that should help in the identification of such artifacts and that we use in this work. Based on Sammon's [42] and Curvilinear Component Analysis' (CCA) [11] loss functions, each of these metrics account for the identification of tears and false neighbors, respectively. The calculation of each metric for a given instance i is given as follows:

$$P_{Sammon}(i) = \sum_j \left(\|\bar{d}_{ij} - d_{ij}\|^2 \times \frac{1}{\bar{d}_{ij}} \right),$$

$$P_{CCA}(i) = \sum_j \left(\|\bar{d}_{ij} - d_{ij}\|^2 \times \frac{1}{d_{ij}} \right),$$

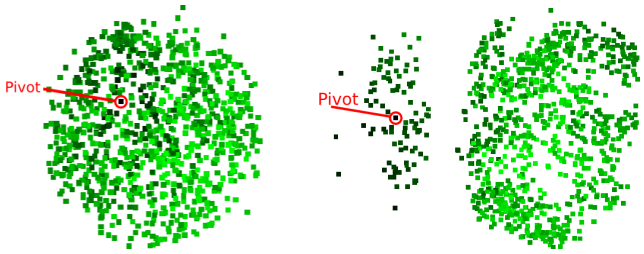
where $\bar{d}_{ij}$ and d_{ij} are the distances between instances i and j in the high- and low-dimensional space, respectively. Each point has an associated P-value (one for Sammon's and other for CCA's loss function), which can be mapped as colors in the projection screen. In our system, the colors vary from black (low error) to red (high error), as seen in Figure 2.



(a) (b)

Figure 2: Colormapping of the (a) Sammon's (tears identification) and (b) CCA's (false neighborhoods identification) erros for the projection of a dataset composed of a 5D-sphere samples.

We also propose the visualization of a distance map related to a projected point called *pivot*. In the projection screen, the user selects a point to be the pivot, and the high-dimensional distance to the pivot is used as a color mapping for each instance. The distance-mapping information can be used as a guidance for the user in the replacement of LAMP control points, in an attempt to reduce the artifacts aforementioned in this section. It can also be valuable for the user decision of using low-dimensional information only or including high-dimensional information in the neighborhood used on iLAMP. In this case, the maps vary from black (small distances) to green (high distances). Figure 3 presents an example of the usage of such information.



(a) (b)

Figure 3: Projection of a 5D sphere dataset. (a) Original Projection colored according to the Distance map of the pivot. It is easy to visualize numerous false neighbors (light green points close to the pivot). (b) Using distance map information, LAMP control points are rearranged. The left points' cloud contains close neighbors to the pivot. Creating new points close to this cloud minimizes the incident of false neighborhoods and tears.

Thus, we cope with false neighborhoods and tears by providing visual feedbacks that allow the user to identify such artifacts. An informed decision regarding the iLAMP-neighborhood type (low- or high-dimensional) can be made in order to reduce or prevent distortions in the iLAMP-generated points.

4 Validation

In this section we present the experiments and results used to measure the quality of the iLAMP method. As previously stated, iLAMP is used to map an instance $\mathbf{p} \in \mathbb{R}^2$ to an instance $\mathbf{q} \in \mathbb{R}^m$ in a coherent way. While there are an infinite number of vectors that may be assigned to $\mathbf{q}$, the goal of iLAMP is to compute the vector that is consistent with the original dataset. More specifically, we compute the vector $\mathbf{q}$ whose multidimensional projection is near to the high-dimensional original surface. In this section we describe the experiments and measurements that lend credence to our back-projection technique.

4.1 Curve Back-Projection

We begin with a qualitative analysis by applying iLAMP to a user designed $2D$ curve back-projecting it into a $3D$ space. Operating in lower dimensions provides visual confirmation of the iLAMP results. In this experiment we construct the *parallel swissroll* dataset by randomly sampling 2000 instances from two adjacent swissrolls that are separated by a small void.

Our parallel swissroll is projected to the plane using LAMP, in which 80 control points are strategically selected along the edges of each swissroll. As demonstrated in Figure 4, a free-hand curve is drawn on the $2D$ visual space, between the two projected swissrolls. Using 60 iLAMP neighbors, we back-project the user defined curve, mapping it back into the original $3D$ space. Figure 4 illustrates the iLAMP reconstructed curve, which appears to be lifted in dimension in a precise and coherent way.

4.2 Hypersphere Reconstruction

The unit hypersphere embedded in an m-dimensional space, $\sum_{i=1}^{m} x_i^2 - 1 = 0$, provides the basis for our quantitative analysis of iLAMP. We create synthetic hypersphere datasets designed to test the robustness of iLAMP under varying sample density and increasing space dimensionality. The datasets are constructed by randomly sampling hyperspheres with different densities (100, 500 and 1000 instances), embedded in multiple spaces (3-, 5-, 10- and 20-dimensional spaces). The different combination of these variables results in the generation of 12 unique hypersphere datasets, against which the following tests are run.

For a given hypersphere dataset, we begin by projecting the samples to the planar domain using LAMP. In these tests, $3\sqrt{n}$ randomly selected samples are selected as the control points used to drive the LAMP projection. Next, 200 points are randomly sampled over the projected domain and iLAMP back-projects them into the original high-dimensional space. Figure 5 illustrates the projection of the 4 hypersphere datasets with 500 samples. In this figure, the red points are the projected samples and the blue points are the 200 randomly selected iLAMP input points.

We additionally note that the iLAMP procedure is applied to each $2D$ point multiple times with different neighborhood sizes. We increment the number of nearest neighbors considered by iLAMP over the interval between 2 and 20, producing 19 different back-projections.

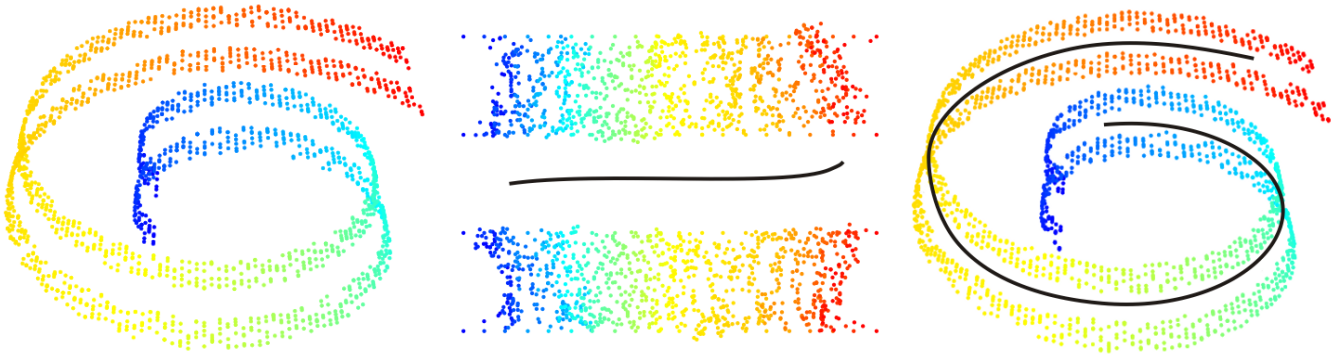
Three metrics monitor the result accuracy, including (1) the distance between back-projected samples and the analytically defined surface; (2) the stress function; and (3) the *LAMP-validation*.

Distance to surface The first measure computes the distance between the back-projected samples, generated via iLAMP, and the nearest point on the unit hypersphere. This value indicates how consistent the iLAMP results are to the originating surface. The distance d_s of the back-projected point $\mathbf{q}$ is defined as,

$$d_s(\mathbf{q}) = |1 - \sum_{i=1}^{m} q_i^2|.$$

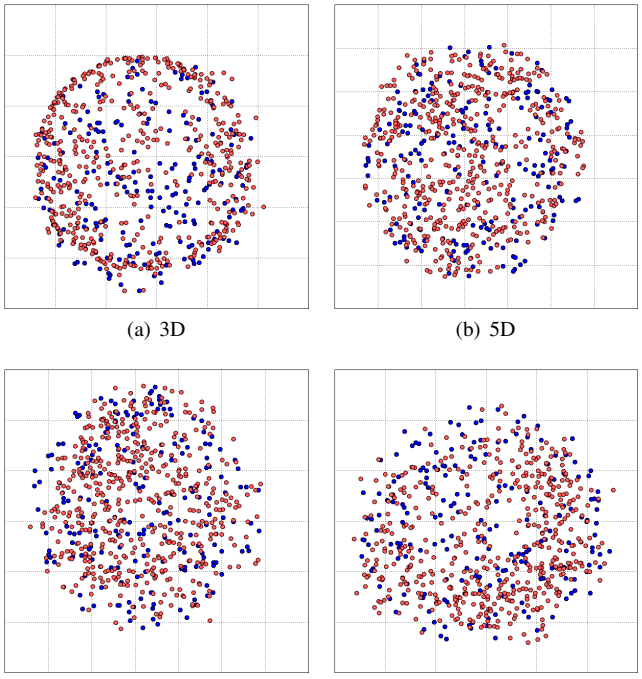
Figure 6 presents box plots of the distances computed between iLAMP's back-projected high-dimensional samples and the hypersphere. Observe that the extrapolated points remain close to the hypersphere surface over which the dataset had been sampled. In particular, the mean distance error is below 0.15 in each experiment. The reported distances rely on the best back-projection solution found amongst the different iLAMP neighborhood sizes used. The following quality measure analyzes the effects of neighborhood size.

Stress function As presented on Section 3.1, the design of iLAMP's transformation matrix is motivated by LAMP. Specifically, the back-projection closely preserves the relative distances between a $2D$ point $\mathbf{p}$ and its neighbors as the relative distances between the



(a) (b) (c)

Figure 4: Swiss roll curve back projection example. (a) original dataset; (b) projection and 2D curve samples (black); (c) 3D swiss roll and back-projected curve.



(a) 3D (b) 5D

(c) 10D (d) 20D

Figure 5: Hypersphere datasets consisting of 500 points (red) in various high-dimensional spaces are projected to the plane using LAMP. 200 new points are randomly sampled over the projection domain (blue) and used for validation of the iLAMP method.

back-projection of $\mathbf{p}$ with the high-dimensional images of its $2D$ neighbors. Projection techniques, such as LAMP, rely on the stress function to measure this preservation of relative distances to validate their dimensionality reduction approaches.

To determine the stress function, let l and n be the number of new and original instances, respectively. Let d_{ij} and $\bar{d}_{ij}$ be the distances between $\mathbf{p}_i$ and $\mathbf{y}_j$, and $\mathbf{q}_i$ and $\mathbf{x}_j$, respectively. Recall that $\mathbf{x}_j \in X \in \mathbb{R}^m$ (the original dataset) and $\mathbf{y}_j = \text{LAMP}(\mathbf{x}_j)$. The iLAMP stress function is defined,

$$s = \frac{\sum_{i=1}^{l}\sum_{j=1}^{n}(d_{ij}-\bar{d}_{ij})^2}{\sum_{i=1}^{l}\sum_{j=1}^{n} d_{ij}^2}.$$

Note that the stress function measures the distance preservation between all pairs of points in the dataset, not just the k nearest neighbors, to measure the global distortion of space.

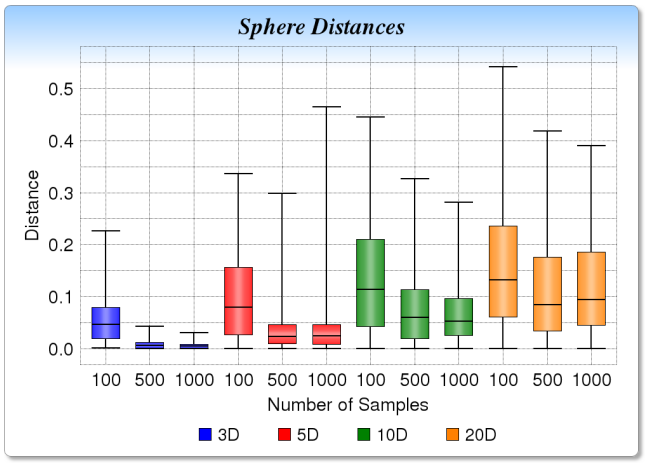


Figure 6: Distances between newly created samples and the sphere surface for each hyper-sphere dataset.

Figure 7 plots the stress function for the iLAMP reconstructions of the multiple hypersphere datasets with respect to the iLAMP neighborhood size (k nearest neighbors). Observe that the number of neighbors is inversely correlated to the projection's stress function; but, with diminishing returns. Further, larger neighborhoods are necessary as the dimensionality of the original dataset increases to maintain a high quality in the back-projection.

LAMP-validation Lastly, the *LAMP-validation* applies the LAMP projection technique to back-projected points, measuring the distance between the new projection and the original point. The user selected $2D$ point $\mathbf{p}$ is lifted into the original high-dimensional space, $\mathbf{q} = \text{iLAMP}(\mathbf{p})$. This point $\mathbf{q}$ is projected back to the $2D$ domain as $\mathbf{p}' = \text{LAMP}(\mathbf{q})$. The *LAMP-validation* measurement becomes,

$$L(\mathbf{p}) = \frac{\|\mathbf{p}-\mathbf{p}'\|}{\|\mathbf{p}'\|}$$

Figure 8 presents the *LAMP-validation* error for various hypersphere datasets of various sample density and dimensionality. This test utilizes the LAMP projection method to attest to the coherence of the extrapolated instances. The small distance residuals suggest that the iLAMP extrapolated instances are consistent with the LAMP projection. In particular, the mean error is below 0.1 across each experiment.

4.3 Experiment Discussion

The results presented in this section indicate that the iLAMP technique is able to extrapolate instances of an existing dataset based on

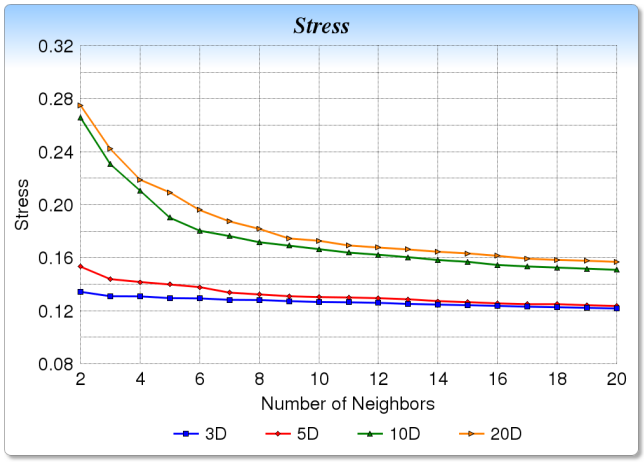


Figure 7: Stress function x number of neighbors (Results obtained with 500 samples' datasets).

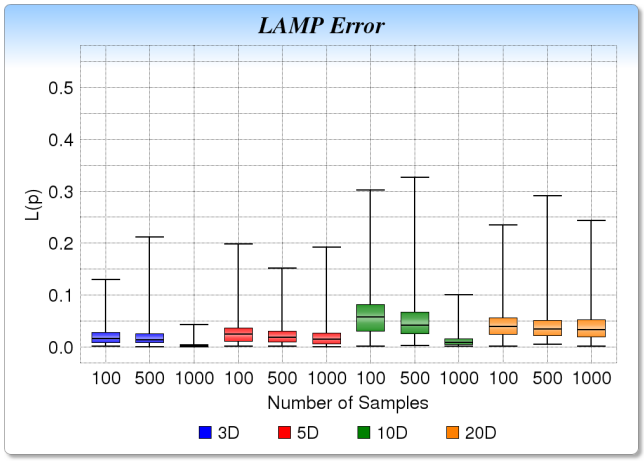


Figure 8: Error measurements for the *LAMP-validation* metric.

local neighborhoods from the layout of a projection. The method creates new instances that are coherent with the original dataset (Figure 6) and its projection (Figure 8). Further, it closely preserves relative distances between point pairs in the m-dimensional space (Figure 7). In the following section we present applications of iLAMP in more specific scenarios.

5 Exploring Parameter Spaces in Optimization Problems

In this section we present an application in which the user expertise is incorporated into the optimization process by iLAMP. In general, optimization problems are solved using automatic methods by either gradient-based or gradient-free techniques. Such techniques start from an *initial guess* and iteratively improve the solution until it gets trapped in a minimum. Gradient-based techniques are very sensitive to the initial guess and different minima may be reached depending on the location of the initial guess. However, creating meaningful starting points for such algorithms is challenging because, in the very beginning, the user may have no idea on where good minimizers are located. Moreover, most optimization problems present non-unique solutions, i.e., there are several satisfactory minimization points. Our application integrates LAMP projection and iLAMP backwards projection in a system that allows the user to explore and inspect by a sampling mechanism regions of interest in the high-dimensional space of possible solutions.

5.1 System

In this application we employ iLAMP to allow the user to interactively explore the high-dimensional optimization space. The application is composed of three main subprograms: 1) LAMP projection visualization method; 2) iLAMP backwards projection and 3) an optimization method. A visualization and interaction window integrates the three moduli while allowing the user to interact the resulting visualization. Figure 9 illustrates the system workflow.

The proposed application receives as input an initial dataset, composed of precomputed local minima, which is projected to the visual space by LAMP (Figure 11-a). LAMP is intrinsically interactive, which allows the user to explore and analyze the initial data by simply manipulating control points (see [25] for details). In this exploration phase, the user can identify regions of interest in the high-dimensional space and resample those regions by adding new points in the visual space. In fact, the system allows the user to create individual points or a set of random points by clicking or drawing rectangles in the visual space (Figure 11-b and c). The new user-defined samples are backward mapped to the multidimensional spaces using our approach. The new points are then inputed as starting points into optimization procedure (Figure 11-d) in order to reveal new local extrema (Figure 11-e). The process can be further refined in specific regions of interest (Figure 11-f) to h) until the user is satisfied with the optimization results.

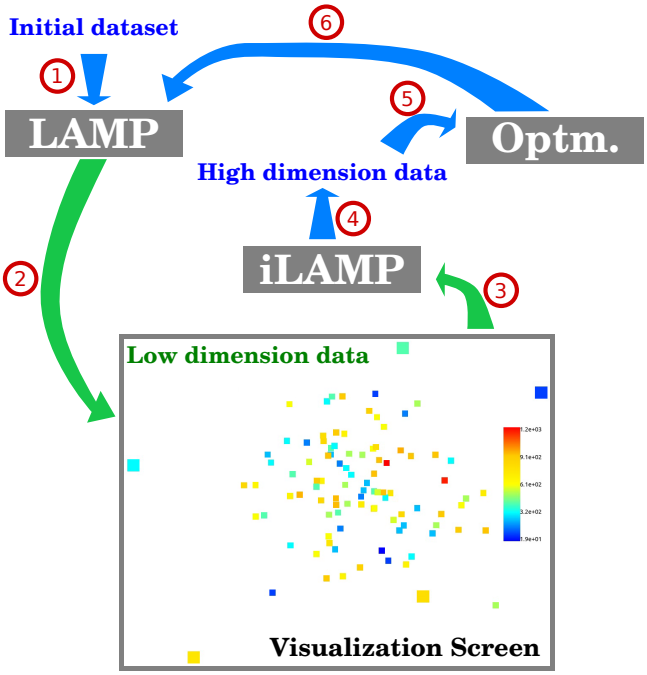


Figure 9: Application workflow; green and blue arrows represent the flux of the low and high-dimensional data, respectively. Instances colored according to optimization function value. (1) Initial data given as input to LAMP; (2) Projection (3) User input passed to iLAMP; (4) New high-dimensional samples; (5) New data is passed as argument to the optimization method; (6) Optimization result incorporated to the dataset.

System in use: In order to provide details about our methodology, we discuss an example step-by-step, demonstrating that our technique may help in the analysis and inspection of optimization spaces.

To illustrate the system we choose the *bird function* as the function to be minimized and the *Levenberg-Marquadt* method to perform the optimization. The bird function is defined as follows:

$$b(\mathbf{x}) = \sum_{i=0}^{(m/2)-1} (sin(x_{i*2}) * e^{(1-cos(x_{i*2+1}))^2} + cos(x_{i*2+1}) * e^{(1-sin(x_{i*2}))^2} + (x_{i*2} - x_{i*2+1})^2 + 106.76),$$

where m is the dimension of the space and x_i the coordinates of point $\mathbf{x}$.

Figure 10 illustrates the bird function for $m = 2$ in the interval $-2\pi < x_i < 2\pi$, where we can clearly see four local minima. If we make $m = 20$ we end up with thousands of local minima, making difficult to figure out where are the best minimizers.

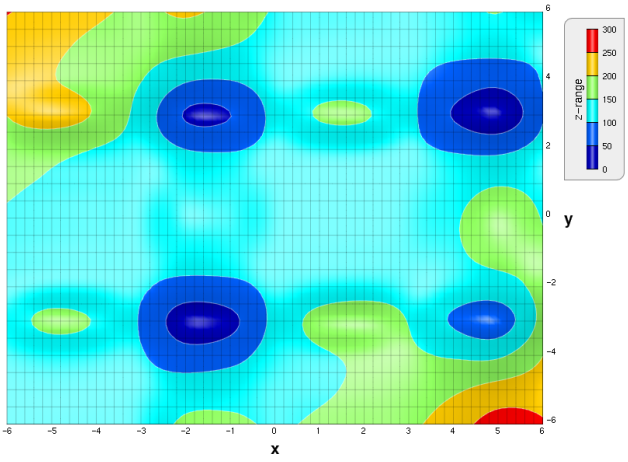


Figure 10: Bird-function for 2 parameters. Four local minima.

We start the 20-dimensional space exploration by running the gradient-based minimization method 100 times with random initial guesses. The smallest minimum contained in the initial dataset has a function value $b(\mathbf{x}) = 19$. The initial dataset was given as input to LAMP, which projected the instances to the visual space (Figure 11-a). Projected samples are colored according to their $b(\mathbf{x})$ function value, which is a key information to guide the user towards regions containing other minima.

With the projection in hand, the optimization space was explored and interactively resampled. iLAMP allows the user to manipulate control points so as to modify the projection and bring out regions of interest where the user can resample by clicking points and drawing rectangular boxes in the visual space (Figure 11-b). Our approach projects the user defined samples back to the high-dimensional space, which are added to the dataset (Figure 11-c). Figure 11-d shows the initial data in green and the new ones created by the user in blue. The new samples are used as input to the gradient-based method which reveals new local minima. Figure 11-e shows the new local minima colored according to their function value. Our exploration system includes some of the new samples as control points for LAMP in order to improve interactivity. A further exploration of the space was performed to bring out more regions of interest (Figure 11-f). Zooming in those regions of interest and repeating the process above a couple of times we end up with approximately 300 new samples (Figure 11-g) which give rise to many extrema, as illustrated in Figure 11-h. In a few seconds we were able to interactively find out minimizer where the bird function is equal to $b(\mathbf{x}) = 1.4e^{-5}$, besides many other local minima that might be of great interest depending on the application.

The example above shows that the proposed system empowers the user with a flexibility not found in other high-dimensional data exploration technique. Indeed, we have tested our approach in optimization functions other than the bird function as well as distinct optimization algorithms. With no exception we could "mine" new minima in a few seconds, always improving the initial results provided by the automated sampling mechanisms built into the optimization softwares, thus making evident the effectiveness of our approach and the importance of adding the user in the process.

6 Conclusion and Future Work

As discussed on Section 3, iLAMP has a very solid mathematical foundation. The accuracy of the proposed technique was attested in qualitative and quantitative experiments, presented on Section 4. Moreover, the application presented on Section 5 indicates that iLAMP can provide a good alternative for including the user knowledge into parameter space exploration of optimization problems, a process usually accomplished with automatic methods.

There are still some points to be improved. One issue is the determination of the number of neighbors k, as the system expects the user to provide this parameter value. Results on Section 4 indicate that k is an important parameter that should be carefully determined in order to have the best iLAMP results. So far we don't have an automatic way to determine such parameter, and we depend on heuristic evaluations to decide this value. Also, it would be most interesting to couple traditional high-dimensional visualization techniques, such as parallel coordinates and scatter plots, with our system. We believe that these techniques could provide a greater insight of each instance represented in the low-dimensional space. Another future work is to investigate how to perform backwards projection using different multidimensional projection techniques, other than LAMP.

Furthermore, a next step in this work is to apply the proposed technique into real-world problems. For instance, the history matching problem in reservoir engineering is already benefiting from the use of multidimensional projection techniques [18] and could be a possible venue to be tackled.

To summarize, in this work we presented the iLAMP method, a novel approach to explore high-dimensional data. The starting point of iLAMP is the 2D projection of a high-dimensional dataset, embedded in a visual space, in which the exploration takes place. iLAMP allows the user to create points and regions in the visual space and map them back into high-dimensional instances, based on the distance of the selected point and the projected data.

Acknowledgements

We would like to thank our colleagues for their useful discussions and advice. We also thank the anonymous reviewers for their careful and valuable comments and suggestions. This research was supported in part by the NSERC / Alberta Innovates Academy (AITF) / Foundation CMG Industrial Research Chair program in Scalable Reservoir Visualization. We also acknowledge Brazilian founding agencies Fapesp e CNPq.

References

[1] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. Lapack: a portable linear algebra library for high-performance computers. In *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, Supercomputing '90, pages 2–11, 1990.

[2] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304–1330, 2007.

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

[4] M. Berthold and L. O. Hall. Visualizing fuzzy points in parallel coordinates. *IEEE Transactions on Fuzzy Systems*, pages 369–374, 2003.

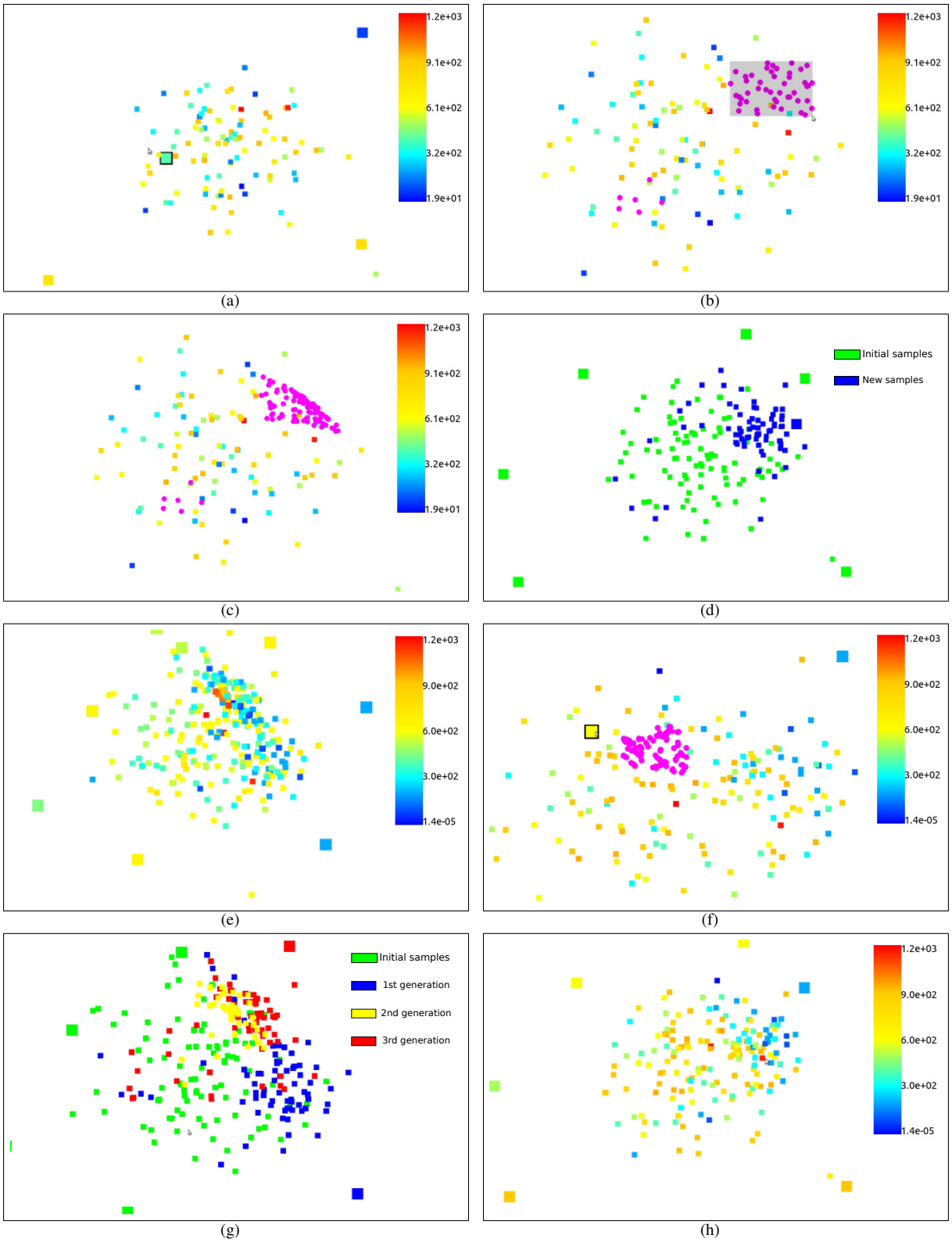


Figure 11: System - (a) Initial dataset projected; (b) New samples created (magenta); (c) New samples incorporated to projection; (d) Green: original dataset, Blue: user-generated samples; (e) After using user-generated samples as input to optimization algorithm; (f) Extrapolating data in other regions; (g) Green: initial dataset; Blue, Red, Yellow: First, second and third user-generated sets; (h) Final layout after optimization of new samples. Color scale on figures (a), (b), (c), (e), (f) and (h) represents the function value of each projected point. Big points are the control points used in the LAMP projection technique. The colors in (d) and (g) are used to differentiate the initial samples from the iLAMP-generated samples. Each color refers to one step in the generation of samples.

[5] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. *LNCS*, 4372:42–53, 2007.

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

[7] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus. *IEEE Transactions on Visualization and Computer Graphics*, 15:1161–1168, 2009.

[8] W. Clevelandand and M. McGill. *Dynamic Graphics for Statistics*. Wadsworth & Brooks/Cole, 1988.

[9] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2nd edition edition, 2000.

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

[11] P. Demartines and J. Herault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, 1997.

[12] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[13] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2008)*, 14(6):1141–1148, 2008.

[14] M. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378 – 394, july-sept. 2003.

[15] Y. Frishman and A. Tal. Multi-level graph layout on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, 13:1310–1319, 2007.

[16] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *IEEE Visualization*, 1999.

[17] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1271–1280, Nov. 2010.

[18] Y. Hajizadeh, E. Amorim, and M. Costa Sousa. Building trust in history matching: the role of multidimensional projection. page to appear, 2012.

[19] W. Harvey and Y. Wang. Generating and exploring a collection of topological landscapes for visualization of scalar-valued functions. *Proc. Symposium on Visualization*, 29, 2010.

[20] J. D. II, E. Anderson, L. Nonato, and C. Silva. Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics*, 16:1560–1568, 2010.

[21] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):249–261, 2009.

[22] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 361–378. IEEE Computer Society Press, 1990.

[23] H. Janicke, M. Bottinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1459–1466, 2008.

[24] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Information Visualization*, pages 125–132, 2005.

[25] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Trans. on Vis. Comp. Graph.*, 17(12):2563 –2571, 2011.

[26] I. Jolliffee. *Principal Component Analysis*. Springer-Verlag, 3rd edition edition, 2002.

[27] F. Jourdan and G. Melancon. Multiscale hybrid mds. In *Proceedings of the Information Visualisation, Eighth International Conference*, pages 388–393, Washington, DC, USA, 2004. IEEE Computer Society.

[28] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, pages 9–12, 2000.

[29] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *ACM Conference on Knowledge Discovery and Data Mining*, pages 107–116, 2001.

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

[31] S. Lespinats and M. Aupetit. False neighbourhoods and tears are the main mapping defaults. how to avoid it? how to exhibit remaining ones? *Quality issues, measures of interestingness and evaluation of data mining models, QIMIE*, pages 55–65, 2009.

[32] A. Morrison, G. Ross, and M. Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *IEEE Symposium on Information Visualization*, page 152, 2002.

[33] A. Morrison, G. Ross, and M. Chalmers. Fast multidimensional scaling through sampling, springs and interpolation. *Information Visualization*, 2(1):68–77, 2003.

[34] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893 –900, 2006.

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

[36] F. Paulovich and R. Minghim. Text map explorer: A tool to create and explore document maps. *Information Visualization*, pages 245–251, 2006.

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

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

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

[40] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. *IEE Symposium on Information Visualization*, pages 89–96, 2004.

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

[42] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 13:401–409, 1964.

[43] T. Schreck, M. Schubler, K. Worm, and F. Zeilfelder. Butterfly plots for visual analysis of large point cloud data. pages 33–40, 2008.

[44] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Information Visualization*, 9:181–193, June 2010.

[45] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[46] D. Swayne, D. Lang, A. Buja, and D. Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

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

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

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

[50] J. Wang, W. Peng, M. Ward, and E. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. *IEEE Symposium on Information Visualization*, pages 105–112, 2003.

[51] L. Yang. Sammon's nonlinear mapping using geodesic distances. In *17th International Conference on Pattern Recognition*, volume 2, pages 303–306, 2004.