An Interactive Visual Testbed System for Dimension Reduction and Clustering of Large-scale High-dimensional Data

Jaegul Choo, Hanseung Lee, Zhicheng Liu, John Stasko, and Haesun Park

Georgia Institute of Technology, USA

ABSTRACT

Many of the modern data sets such as text and image data can be represented in high-dimensional vector spaces and have benefited from computational methods that utilize advanced computational methods. Visual analytics approaches have contributed greatly to data understanding and analysis due to their capability of leveraging humans' ability for quick visual perception. However, visual analytics targeting large-scale data such as text and image data has been challenging due to the limited screen space in terms of both the numbers of data points and features to represent. Among various computational methods supporting visual analytics, dimension reduction and clustering have played essential roles by reducing these numbers in an intelligent way to visually manageable sizes. Given numerous dimension reduction and clustering methods available, however, the decision on the choice of algorithms and their parameters becomes difficult. In this paper, we present an interactive visual testbed system for dimension reduction and clustering in a large-scale high-dimensional data analysis. The testbed system enables users to apply various dimension reduction and clustering methods with different settings, visually compare the results from different algorithmic methods to obtain rich knowledge for the data and tasks at hand, and eventually choose the most appropriate path for a collection of algorithms and parameters. Using various data sets such as documents, images, and others that are already encoded in vectors, we demonstrate how the testbed system can support these tasks.

Keywords: clustering, dimension reduction, high-dimensional data, visual knowledge discovery

1. INTRODUCTION

The volume of available data has been increasing at an exponential speed in recent years. Many of the modern data are generated in various forms such as documents and images of which the raw data can be represented in a high-dimensional vector space, allowing various computational methods to be applied. For instance, text documents can be encoded using a bag-of-words model, and images are represented using their feature point descriptors,[1] resulting in hundreds of thousands of dimensions.

Given high-dimensional data, understanding and analyzing them become more challenging. Visual analytics[2,3] has gained increasing interest due to its capability of leveraging humans' ability of quick visual insight in data analyses and decision processes. However, many state-of-the-art visual analytics techniques or systems are not equipped for high-dimensional large-scale data. One of the reasons is that although humans are good at visually grasping an overall structure, when the number of visualized objects becomes large, it is often difficult to extract meaningful information from visualization. Another factor is the limited dimension of a screen space where high-dimensional data have to be visualized. For instance, parallel coordinates, a widely-used visualization technique for multi-dimensional data, do not scale well even when the dimension reaches several tens.

To improve this scalability issue, computational methods can support visual analytics by transforming the original data into a more compact and meaningful representation. Among various methods, two main ones, dimension reduction and clustering, play an essential role in visual analytics of large-scale high-dimensional data owing to their nature to reduce the numbers of features and data items into manageable sizes, respectively.

Further author information: (Send correspondence to J.C.)
J.C., J.S., H.P.: E-mail: {jaegul.choo, stasko, hpark}@cc.gatech.edu
H.L., Z.L.: E-mail: {hanseung.lee, zcliu6}@gatech.edu

Dimension reduction methods can reveal meaningful information by allowing the visual representation of high-dimensional data in a much lower-dimensional space. In addition, it allows visualization of high-dimensional data in the form of a 2D/3D scatter plot in which one can obtain insight about data relationships with respect to the geometric locations of data. On the other hand, clustering provides an overview of large-scale data in terms of a small number of groups based on their semantic coherences. Such cluster information can then guide us to a proper data group of interest on which we can further focus our analysis.

Given a wide variety of computational methods including dimension reduction and clustering methods, however, it is not easy to determine which method to choose and how to use it properly for a certain data set and a certain task. Sometimes, when a specific method is used for a certain data set, its performance may be dependent on how the data is pre-processed beforehand. In addition, many modern computational methods often require decisions on multiple parameters. Yet there is no theoretical guideline for an optimal set of parameters for a given problem, and we have to go through multiple trials only to obtain some initial understanding of parameter values. As the algorithm gets more complicated, it becomes more difficult for users to understand what these parameters mean and how to select them properly. Consequently, many visual analytics systems choose a certain computational method, which is often basic and/or generic, and treat it as a black box with fixed parameter values while focusing on the subsequent analysis after obtaining the output from it. However, without an appropriate choice of algorithms and their parameters, the performance of these methods may not be satisfactory enough to start an analysis with.

Due to these difficulties, the current state of the art in visual analytics has not taken full advantage of the recent advancements of computational methods. To tackle this problem, we claim that users have to be provided with the capability of interactively trying out various computational methods and their parameters and reviewing their results at a visual level without having to know the details of algorithms. As a cornerstone to achieve this claim, this paper presents an *interactive visual testbed system for dimension reduction and clustering*, the two essential computational methods for the visual analytics of large-scale high-dimensional data.

The main contributions of the proposed testbed system are as follows. First of all, given various types of input data such as text documents, images, and vector-encoded data, the testbed system provides extensive capabilities to interactively select data pre-processing options and choose a wide variety of clustering and dimension reduction methods along with their parameters. The output of these processes are then visualized in several forms, e.g., parallel coordinates and a scatter plot, equipped with various interaction capabilities, e.g., accessing the original data items and brushing and linking between multiple views. Additionally, the testbed system facilitates easy comparisons between different dimension reduction and clustering results by computationally aligning them. Finally, the testbed system is implemented in a highly modular way so that new data types and dimension reduction/clustering methods can be easily integrated to the current system.

Note that even though the testbed system can be used by anyone who wants to apply various methods to their own data, some background knowledge about machine learning and data mining would be of great help in fully utilizing the testbed system via understanding the data and the applied methods simultaneously. For example, machine learning researchers/developers, who wish to easily plug in and visually evaluate their own methods in practical data analysis scenarios, would be able to receive significant benefit from the testbed system.

The rest of this paper is organized as follows. In Section 2, we review the relevant literature in terms of dimension reduction and clustering methods as well as the visual analytics systems adopting them. Section 3 describes the details of the testbed system as well as several main computational methods used in the system. Section 4 shows various usage scenarios of the testbed system, and finally Section 5 presents conclusions along with future work.

2. RELATED WORK

In this Section, various dimension reduction and clustering methods applicable to visualization are first reviewed. Afterwards, we discuss some of the currently available visual analytics systems that adopt these computational methods.

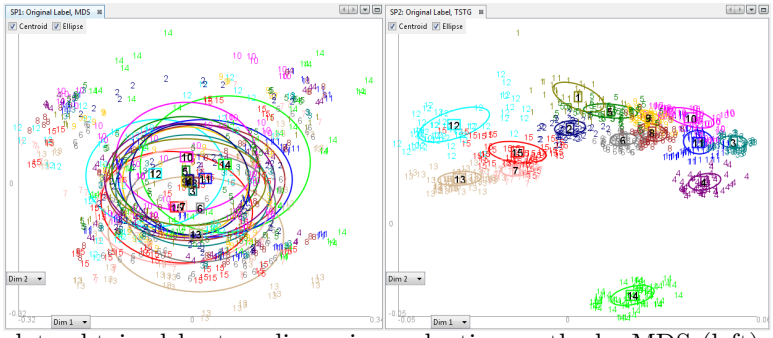


Figure 1: 2D Scatter plots obtained by two dimension reduction methods, MDS (left) and LDA (right), for a facial image data set. A different color corresponds to a different cluster.

2.1 Dimension Reduction and Clustering for Visualization

Dimension reduction has long been one of the main research topics in data mining and statistical machine learning areas. Numerous dimension reduction methods have been proposed, among which the most commonly used dimension reduction methods include principal component analysis (PCA),[4] multidimensional scaling (MDS),[5] and linear discriminant analysis (LDA).[6–8]

In addition to traditional data analysis problems, they have also been widely utilized in visualization due to their capability of representing high-dimensional data as a form of scatter plots in 2D/3D space. In a scatter plot, each data item is represented as a point and its 2D/3D coordinate is determined from the dimension-reduced representation. In general, the relative locations among data points reflect the pairwise relationships or proximities among data items.

Each dimension reduction method has its own optimization criteria and behaviors, which result in different visualizations. For instance, the recently proposed manifold learning algorithms, e.g., isometric feature mapping (ISOMAP),[9] locally linear embedding (LLE),[10] and Laplacian Eigenmaps (LE),[11] try to preserve the relationships between the local neighborhood rather than global relationships. These methods have been successfully applied to the data that originally have a low-dimensional manifold structure, and often they demonstrated their capability to reveal such a manifold structure in 2D/3D visualizations. However, most of these methods present just several visualization snapshots of limited data sets with no interaction abilities.

Another aspect to consider when applying dimension reduction in visualization applications is the cluster structure of data. A majority of dimension reduction methods take into account only the pairwise relationships between data items. In practice, however, it is not easy to obtain much insight from the 2D/3D scatter plot generated by them for a large number of data items. The left figure in Figure 1 is a visualization example of such a dimension reduction method, MDS, for a facial image data set. Let us, for now, ignore the colors, which indicate the cluster labels. This visualization shows most of the data as a single chunk with a few outliers placed outside. Although these points may give some interesting insight about why they appear to be outliers, one cannot get much more information from this visualization.

Another type of dimension reduction methods incorporates additional information about the cluster structure of data in addition to individual data items. Since these dimension reduction methods require the assigned cluster label associated with each data item as an input, they are called supervised dimension reduction methods while the previous methods are called unsupervised dimension reduction methods. Some representative supervised methods include LDA[6] and orthogonal centroid method (OCM).[12] The right figure in Figure 1 is an example of LDA visualization. This figure visualizes the data as groups of items computed by LDA based on the given cluster labels, and one can obtain better insight about the overall data structure at the cluster level over the individual data level.

Representing the cluster structure has been one of the main concerns in many studies on dimension reduction and 2D/3D scatter plot visualizations even when unsupervised dimension reduction methods are used. Many methods have been evaluated regarding their ability to visualize cluster structures, which are hidden at the time

of computing dimension reduction. For instance, a recently proposed dimension reduction method, t-distributed stochastic neighborhood embedding (t-SNE),[13] shows its capability of grouping data and revealing the true cluster structure in 2D scatter plot visualizations.

Given the importance of the cluster structure in large-scale data visualization, clustering methods can add a significant value to visual analytics approaches by enabling visual understanding of the overview of data. Clustering partitions the entire data into groups or clusters so that the data items in the same cluster are more similar to each other than to those in different clusters. The resulting grouping information is in a form of cluster labels, which act as an additional categorical variable associated with data items. Such cluster information can be color-coded in visualization, as shown in Figure 1, and help us understand the cluster structure in the data clearly in visualization.

Clustering, along with dimension reduction, has also been one of the well-studied research topics in data mining and machine learning areas. Widely-used methods include k-means clustering, spectral clustering,[14] and Gaussian mixture models. Recently, more advanced methods such as non-negative matrix factorization (NMF)[15] and latent Dirichlet allocation[16] have shown their successful applications in image segmentation and document topic modeling, etc. These methods are usually evaluated using the data set whose cluster label information is already known and by comparing between the true cluster labels with those obtained by the computational method. However, given the data set that may not have a clear cluster structure, clustering is typically a very challenging task, and thus it is often the case that the resulting cluster quality is unsatisfactory. From a visual analytics perspective, unsatisfactory clustering makes it difficult to understand the coherent meaning of each cluster and how one cluster contrasts with another. For instance, in recent applications of latent Dirichlet allocation for document topic modeling, while several coherent topic clusters have been successfully revealed for the document data, many other topics often seem unclear to understand.

Even with the obvious needs of computational methods such as dimension reduction and clustering in visual analytics, various issues such as data noise and improper algorithm and parameter choices, as described in Section 1, prevent their initial results from being practically useful enough to support the subsequent visual analysis. Nonetheless, among data mining and machine learning communities, which supply supposedly better computational methods, the efforts of interactively improving these initial results in real-world data analysis seem to be overlooked.

2.2 Visual Analytic Systems using Dimension Reduction and Clustering

In information visualization and visual analytics communities, various visual analytics systems incorporating computational methods such as dimension reduction and clustering have been proposed to deal with large-scale high-dimensional data. In this section, several systems such as IN-SPIRE,[17] Jigsaw,[18] GGobi,[19] iPCA,[20] and WEKA[21] are discussed.

IN-SPIRE[17] is one of the well-known visual analytics systems for document data in which dimension reduction and clustering play main roles. Given a set of documents, IN-SPIRE first encodes them as high-dimensional vectors using a bag-of-words model. Then it applies k-means clustering with a pre-defined number of clusters. PCA is computed on cluster centroids and applied to the entire data, which gives 2D coordinates of document data. Based on these 2D coordinates, the scatter plot called a galaxy view is shown to users with a keyword summary for each cluster placed at the cluster centroid. Owing to the simple algorithms adopted such as PCA and k-means, IN-SPIRE can deal with a fairly large amount of data, but it provides only a limited number of interaction capabilities to change the algorithms and their settings.

Jigsaw[18] is another well-known visual analytics system for document analysis. The main information that Jigsaw utilizes for visualization is named entities such as person name and location and their co-occurrences between documents. Automatic named-entity extraction is one of the key computational components in their analysis. The named-entity extraction can be viewed as dimension reduction that reduces the number of keywords out of the entire vocabulary. Users can modify the list of named-entities by manually adding/removing them. Jigsaw also provides a cluster view by using the k-means algorithm and visualize the resulting clusters as groups of documents as well as their keyword summary. Jigsaw also supports basic interactions with clustering such as changing the number of clusters and providing seed documents.

GGobi[19] is an interactive visualization system for high-dimensional data that are already encoded. It mainly uses a 2D scatter plot, where the two dimensions are generated by grand tour.[22] The difference of grand tour from other dimension reduction methods is that it provides an interaction to explore the high-dimensional space by continuously changing the basis vectors that data items are projected into. However, the grand tour method is applicable only when the data dimension is not significantly high, and thus its application is limited when dealing with hundreds or thousands of dimensions, which is often the case in many data types such as text documents, images, and bio data.

Another system, iPCA,[20] which also takes high-dimensional data as an input, utilizes PCA as the main visualization technique. One of the main advantages of iPCA is that beyond 2D/3D scatter plots, it visualizes the reduced-dimensional data in a higher dimension than 2D or 3D via parallel coordinates. In generally, dimension reduction from the original high-dimensional space to 2D/3D space introduces significant information loss. In iPCA, it follows a useful idea to reduce the data dimension to an intermediate one that can be visualized without much clutter via parallel coordinates and then to interact with these intermediate dimensions to obtain particular scatter plots. Another aspect of iPCA is that it visualizes the PCA basis vectors in addition to the data items. In doing so, users can understand the role of the reduced dimensions in their visualizations, which leads to a better understanding about the data set as well. Even with these advantages, however, iPCA cannot handle very high-dimensional data since iPCA visualizes each of the original dimensions.

Finally, WEKA[21] is mainly a library of various machine learning algorithms for high-dimensional data with several interaction capabilities. Various algorithms can be applied to data, and their performances can be evaluated based on various measures. In addition, WEKA provides simple types of visualizations such as histograms, scatter plots, etc. Although WEKA is similar to our testbed system in that it provides flexible algorithm choices and settings, most of its visualizations and interactions are focused on the used methods rather than data exploration. For example, WEKA does not support any interactions from its visualizations such as filtering operations and raw data access.

As discussed above, most of the current visual analytics systems do not fully utilize a wide variety of computational methods. They adopt generic traditional methods for a broad applicability to various data sets and/or treat computational methods as a black box with little options to control them, which would hamper the interactive visual analysis. In this respect, the testbed system provides the unique capability of bringing a variety of algorithms along with full control to practical visual analytics scenarios.

3. TESTBED SYSTEM

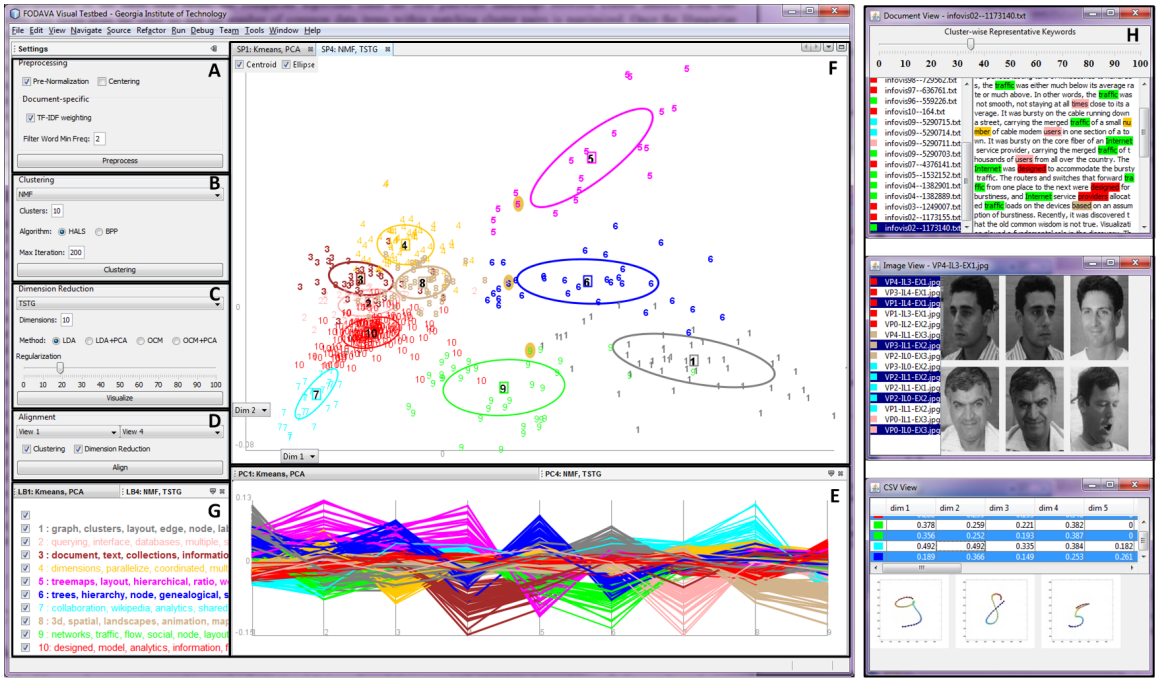
In this Section, we describe the testbed system* in detail. First, in Section 3.1, we introduce the modules in the system and explain how the overall system works. Next, we describe the details of each module from both the computational and the interactive visualization points of view in Sections 3.2 and 3.3, respectively. Finally, in Section 3.4, we discuss implementation details of the system and how the current system can be extended to adopt new data types and clustering/dimension reduction methods.

3.1 Basic Workflow

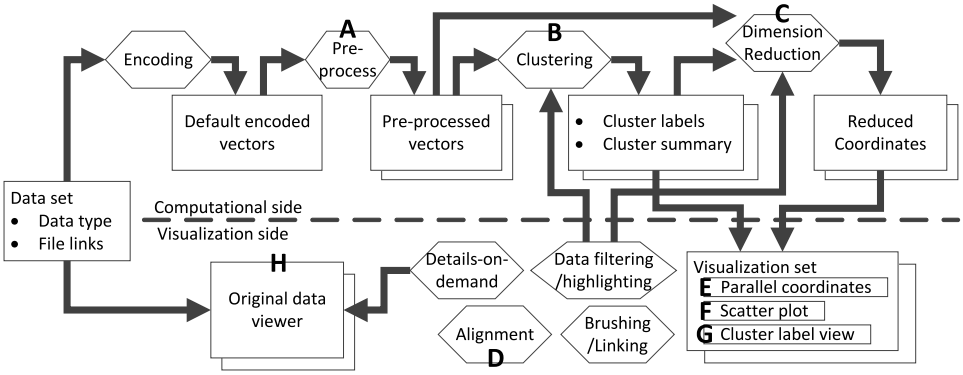
As shown in Figure 2, the testbed system mainly has two parts: the computational and the interactive visualization parts. At the computational part, the testbed system is composed of 1. vector encoding, 2. pre-processing, 3. clustering, and 4. dimension reduction. At the interactive visualization part, the testbed provides the following interactive visualization modules: 1. parallel coordinates, 2. the scatter plot, 3. the cluster label view, and 4. the original data viewer.

The basic workflow of the testbed system is as follows. Once a data set is loaded, data items are represented as high-dimensional vectors via a default encoding scheme. Then, users can interactively change the options for pre-processing, clustering, and dimension reduction methods. Each specification of these three components instantiates a particular visualization set composed of the parallel coordinates view, the scatter plot view, and the cluster label view. To generate these views, the output of dimension reduction, i.e., reduced-dimensional

*An introductory video can be downloaded at `http://fodava.gatech.edu/files/testbed-software/testbed.mp4`, and the executable files with the used data sets are available at `http://fodava.gatech.edu/fodava-testbed-software`.



(a) The system overview.



(b) The general workflow. Hexagonal blocks correspond to operations/interactions, and rectangular ones to operation inputs/outputs or visualization modules. Stacked rectangles indicate their multiple instantiations, which are dynamically maintained by the system.

Figure 2: The overview and the workflow of the system. User interfaces for pre-processing (A), clustering (B), dimension reduction (C), and alignment (D) are available. Lower-dimensional data from dimension reduction are visualized as parallel coordinates (E), and the two selected dimensions are shown in the scatter plot (F). Cluster indices/summaries (G) are shown, and the original data can be accessed (H).

representation, acts as the coordinates of data items in the parallel coordinates view, and two user-selected dimensions of this view are visualized in the scatter plot view. In all three views, the output of clustering, i.e., grouping information of data items, is color-coded along with the cluster name/summary provided in the cluster label view.

The testbed system can generate as many visualization sets as needed depending on different specifications of dimension reduction and clustering, and users can explore a certain visualization set and compare between different visualization sets. To facilitate an easy comparison between different visualization results, the testbed system offers the capability of aligning the different clustering and dimension reduction outputs. In addition, users can highlight and/or filter out certain clusters/data items and look into the details of the selected data

items in the original data viewer. Users can also apply another set of clustering and/or dimension reduction to the selected data items to create new visualization sets.

3.2 Computational Modules

3.2.1 Vector Encoding

The testbed system can take various types of data such as text documents, images, and pre-encoded vectors in a comma-separated-values (CSV) file format. For document and image data, the testbed system provides built-in vector encoding modules. For instance, the testbed system supports bag-of-words encoding for document data in a sparse matrix form with stop word removal and stemming. Image data are converted into vectors of rasterized gray-scale pixel values. The high-dimensional vectors obtained in this stage act as initial default vectors on which the following pre-processing is performed.

3.2.2 Pre-processing

Once the default vectors are generated, the system shows pre-processing options depending on the data type (Figure 2A). The following options are provided in common for all data types: 1. normalization, which scales data vectors so that their norms equal to one, and 2. centering, which translates data vectors so that their empirical mean is zero.

In addition, for text documents, we provide options of 1. removing the terms appearing in less than a user-specified number and 2. applying the term-frequency-inverse-document-frequency (TF-IDF) weighting scheme. For images, available are the following options: 1. reducing image sizes to a user-specified ratio to enhance the computational efficiency and 2. applying contrast limited adaptive histogram equalization.[23]

The testbed system maintains multiple instances of different pre-processed vector sets, and users can interactively generate and/or choose one of them and proceed to perform its clustering and dimension reduction.

3.2.3 Clustering

Given the default or pre-processed set of high-dimensional vectors, the clustering module performs a user-selected clustering method with specified options (Figure 2B), which assigns each data item a cluster label. The testbed system currently provides the following clustering methods: 1. k-means, 2. agglomerative hierarchical clustering,[24] 3. Gaussian mixture models, and 4. NMF. Once a specific method is selected in the system, user interfaces to specify the number of clusters as well as method-specific parameters are dynamically shown with their suggested default values (Figure 2B).

Additionally, when a data set has pre-given labels, the clustering method list includes an additional item called 'Use original labels' so that users can explore data with respect to the pre-given labels.

3.2.4 Dimension Reduction

Given the high-dimensional vector representations of data items, the dimension reduction module reduces the data dimension from possibly hundreds of thousands to a visually manageable size, which makes it possible to visualize the data in forms of parallel coordinates and/or a scatter plot. The testbed system provides both supervised and unsupervised dimension reduction methods, as discussed in Section 2.1. In cases of supervised methods, the cluster label, which is an additional required input to run dimension reduction, is taken from the output of the clustering module.

The currently available dimension reduction methods in the system include supervised ones such as 1. LDA, 2. OCM, 3. centroid method (CM),[12] 4. two-stage methods (TSTG),[25] 5. discriminative neighborhood metric learning (DNML),[26] and 6. kernel LDA,[27] and unsupervised ones such as 7. PCA, 8. metric and nonmetric MDS, 9. Sammon mapping,[28] 10. ISOMAP, 11. LLE, 12. local tangent space alignment (LTSA),[29] 13. maximum variance unfolding (MVU),[30] 14. LE, 15. diffusion maps (DM),[31] 16. t-SNE, and 17. Kernel PCA.[32] Similar to the clustering module, once a specific method is selected in the system, user interfaces to specify the number of reduced dimensions as well as method-specific parameters are dynamically shown along with their suggested default values (Figure 2C).

3.3 Interactive Visualization Modules

3.3.1 Parallel Coordinates View

Given an output from the computational part, i.e., lower-dimensional representations of data items and their cluster labels, the testbed system takes a natural way to visualize the lower-dimensional data in parallel coordinates with a color coding based on the cluster labels (Figure 2E). In this view, the testbed system supports zoom-in/out via mouse wheel scroll and data selection via mouse drag-and-drop.

3.3.2 Scatter Plot View

Although parallel coordinates can fully visualize an output from the computational part, this view is often ineffective for humans to perceive the relationships between data items, and it does not scale well in terms of the number of data items and dimensions since each line representing a single data item occupies numerous pixels in a screen space. Due to these limitations, the testbed system visualizes data in a 2D scatter plot (Figure 2F) by selecting two of the parallel coordinates dimensions with the same color encoding as in the parallel coordinates view.

In the scatter plot view, users can interactively change these dimensions corresponding to horizontal and vertical axes via combo boxes shown in the lower left part of the view. In addition, the testbed system shows cluster centroids and ellipses, which summarize how the data within each class are distributed, and these features can be turned on/off via check boxes shown in the upper left part of the view. Similar to the parallel coordinates view, supported are zoom-in/out via mouse wheel scroll and data selection via mouse drag-and-drop. Once a subset of data is selected, users can apply another clustering/dimension reduction only on the selected data items.

3.3.3 Cluster Label View

The cluster label view (Figure 2G) shows the cluster index, color, and summary in a simple list form. Currently, the cluster summary is provided only for the text documents type, which is the most frequent keywords in each cluster. Upon clicking a certain cluster index or summary, the corresponding data items are highlighted with thicker lines/points in the parallel coordinates and the scatter plot views. Unchecking the checkboxes, which are shown in the left side of the view, hides the corresponding data items in the two views

3.3.4 Accessing Original Data

Both in the parallel coordinates and the scatter plot view, users can access the original form of data for user-selected data items/clusters in the original data viewer. Currently, the testbed system provides three different original data viewers depending on the data type, e.g., text documents, images, and pre-encoded vectors (Figure 2H).

In all the original data viewers, selected data items are shown as a list with their cluster colors on the left, and users can multi-select items in the original data viewer to see the original data items. These selected items are then highlighted with dark yellow marks in the scatter plot view (Figure 2H). For text documents, adopting the idea in,[33] we color-code a user-specified number of representative keywords per each cluster with the corresponding cluster color, which helps in understanding why a document belongs to a certain cluster. For pre-encoded vectors, if these vectors are associated with another type of data, these data can also be accessed as shown in the third viewer in Figure 2H.

3.3.5 Supporting Multi-view Exploration

Once the 'visualize' button is clicked (Figure 2C) after specifying computational methods, i.e., pre-processing, clustering, and dimension reduction, a new set of the parallel coordinates, the scatter plot, and the cluster label views are instantiated. Each of these three views is created as an individual tab in its corresponding location, and multiple views are maintained flexibly in the testbed system, as shown in Figure 2(a). For example, any view can be popped out as an independent window and/or split horizontally/vertically in order to make it easy to compare between different sets of views due to different computational methods. When a certain view is activated by a mouse click, all the options of pre-processing, clustering, and dimension reduction used to generate the view are shown in the left in Figure 2(a).

Between different views with such a flexible layout, the testbed system supports a brushing-and-linking capability. In the current testbed system, if certain data items/custers are selected in one view, the corresponding data items in all the other views are highlighted as well. We use different colors for highlighting depending on whether the highlighted data items are due to the same view or a different view, which helps identifying the source view in which the data selection was made.

3.3.6 Aligning Different Views

In addition to the above-described multi-view management and brushing-and-linking capability, the testbed system provides a more active means to facilitate easy comparison between visualization sets composed of different clustering and dimension reduction results. To be specific, for a user-selected pair of visualization sets, users can align the clustering and/or dimension reduction outputs (Figure 2D), which are then reflected to visualization sets.

To align the two different clustering results, the testbed system performs the Hungarian algorithm.[34] Given two different cluster assignments of the same data items, the Hungarian algorithm finds the best pairwise matchings between their cluster indices so that the number of common data items within matching cluster pairs is maximized. Once the Hungarian algorithm finishes, the testbed system changes the cluster indices and colors of the second visualization set according to the matching clusters of the first visualization set. As a result, users can maintain the consistent cluster indices/colors when comparing the two given visualization sets.

On the other hand, the testbed system handles the alignment of dimension reduction results via Procrustes analysis.[35,36] Although there exist many advanced methods to align the two sets of vectors,[37] we chose Procrustes analysis due to its computational efficiency. Procrustes analysis transforms the second set of vectors via a rigid transformation, which allows only translation, rotation, and reflection, so that their Euclidean distances to the corresponding data vectors in the first set are minimized. Currently, instead of aligning the entire dimensions, the testbed aligns only the two dimensions selected in the scatter plot view so that the alignment between the two scatter plot views are maximized. These alignment functionalities help users understand how differently the corresponding data items/clusters are placed between the two scatter plot views.

3.4 Implementation and Extensibility

The current testbed system is mainly implemented in JAVA to achieve various GUI and interaction capabilities. In order to support flexible window management, NetBeans Rich Client Platform and IDE[†] are used.

Most of the internal computational methods are, however, written in MATLAB. There are several reasons of using MATLAB codes instead of porting them to JAVA. First of all, in many cases, the source codes of advanced computational methods are readily available in MATLAB due to its simplicity for matrix computations. In this respect, it would be burdensome to re-implement each of these methods in a different programming language in order to make them visual and interactive, and it will eventually become difficult to keep up with the pace of new technologies.

Furthermore, MATLAB provides highly optimized matrix computations. For instance, MATLAB, by default, auto-identifies the parallelizable subroutines in the code and runs full CPU cores even in a single PC. There also exist many efficient mature core computational methods. For example, the k-means function in MATLAB provides various options for a distance metric to be used (Euclidean, city block, cosine, and correlation), and a seed initialization (random, uniform, pilot-clustered, and user-selected seeds). Due to these reasons, the current testbed system interface with the computational methods via a custom JAVA library file created by MATLAB.[‡]

In terms of the extensibility of the testbed system, we designed it in a completely modular way so that it can easily accept new data types and clustering/dimension reduction methods. For instance, if one wants to use the testbed system for a speech data type, one needs to implement only the encoding module, the possible pre-processing options specific to the speech data type, and the original data viewer that can play audio data. Otherwise, by performing vector encoding separately and putting the encoded vectors as an input to the system, one can easily utilize the full capability of the testbed.

[†]http://netbeans.org/features/platform/index.html

[‡]http://www.mathworks.com/products/javabuilder/

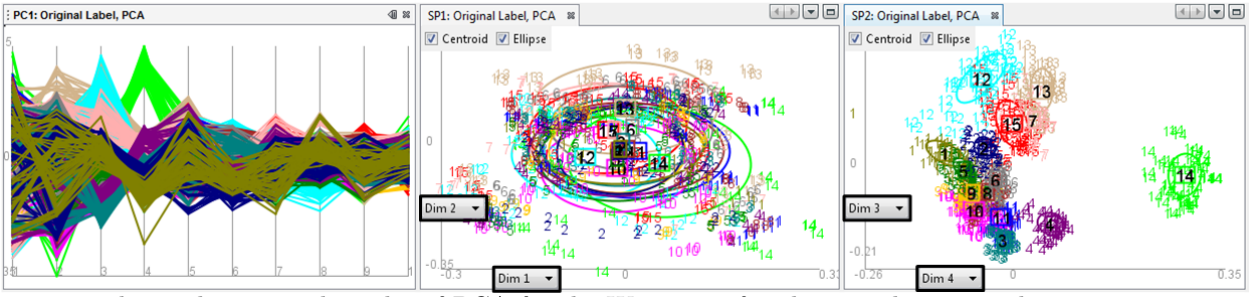
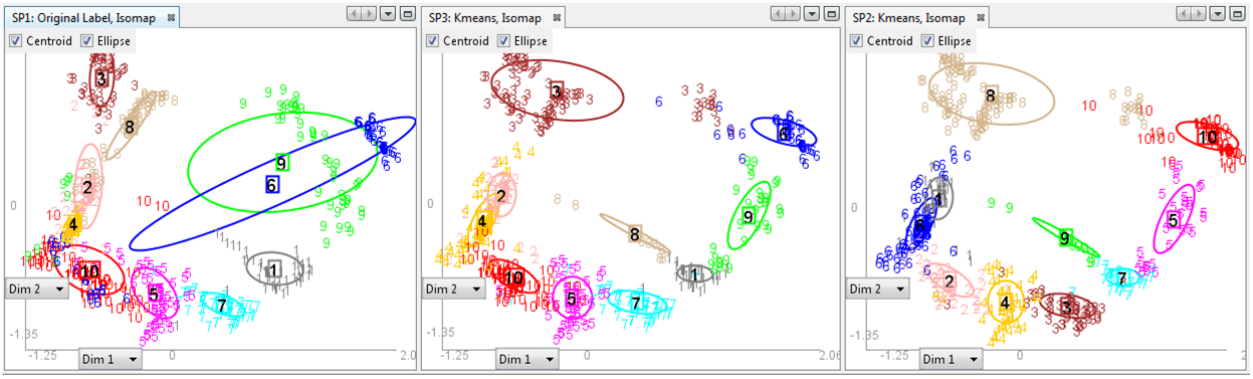
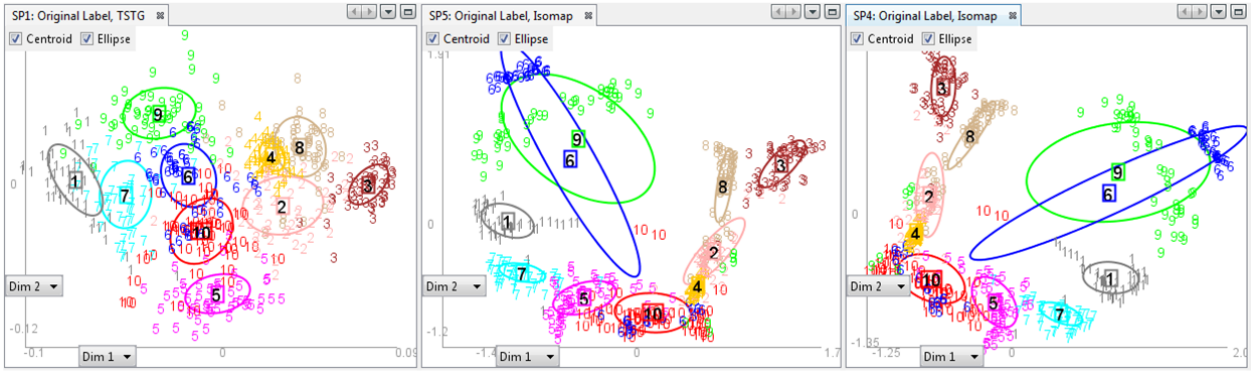


Figure 3: The 10-dimensional results of PCA for the Weizmann facial image data set. The pre-given person id was used as a color label. The first figure is the parallel coordinates of the entire 10-dimensional representations, and the second and the third are the scatter plots of (1, 2)- and (3, 4)-dimensions, respectively.



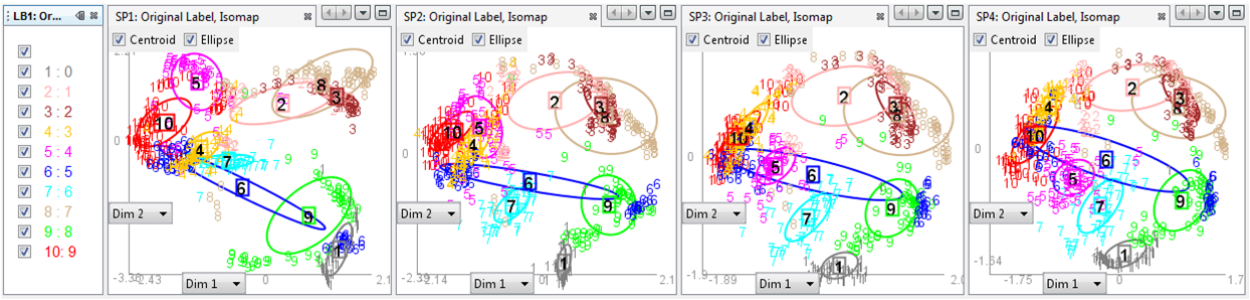
(a) The alignment of clustering. For the Pendigits data set, the first figure uses the original cluster labels, and the other two uses the same cluster labels generated by k-means. In all three figures, ISOMAP is used with the same parameter values.



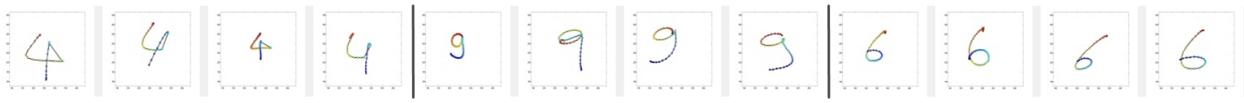
(b) The alignment of dimension reduction. For the Pendigits data set, the first figure uses TSTG and the other two use ISOMAP with the same parameter values. In all three figures, the original cluster labels are used.

Figure 4: The effects of alignment. In both (a) and (b), the first is the reference scatter plot view for alignment, and the second is the aligned plot of the third while the third is an un-aligned one.

Adding new dimension reduction/clustering methods is also a simple process. Currently, the implementation of each computational method is composed of two source code files. One file performs the computation by taking an input and generating an output as a primitive two-dimensional double array type, and the other is for user interfaces to change the method-specific options. Therefore, whether the implementation of a new method is written in MATLAB or JAVA, as long as it deals with two-dimensional double array type as an input and an output, it can be easily integrated into the current testbed system without having to modify the entire system.



(a) The scatter plots for the Pendigits data set generated by ISOMAP with different parameter values, k =12, 20, 30, and 50, respectively. The cluster labels represent the digits of data items, as shown on the left. The three figures on the right are aligned plots with respect to the first. As the parameter increases, the cluster '5' (the digit '4'), moves from the top left near the cluster '10' (the digit '9') towards the cluster '7' (the digit '6').



(b) The sample data for the digits '4', '9', and '6.' Note that the vector representation of the Pendigit data set encodes the pen trace coordinates, which start at the red color and end at the blue in these samples.

Figure 5: The effects of a parameter change in ISOMAP.

4. USAGE SCENARIOS

4.1 Data Sets

To show how the testbed system can be utilized in various visual analytics scenarios, we use three different data sets: 1. Pendigits (pre-encoded vectors), 2. Weizmann (images), and 3. InfoVisVAST (text documents).

The Pendigits data set[§] is composed of 10,992 handwritten digit data items, each of which is a 16-dimensional vector representing pen trace coordinates.[38] The data set has 10 clusters in terms of which digit each data item corresponds to, i.e., '0', '1', ..., and '9.' For our experiments, 50 data items have been chosen from each cluster, resulting in 500 items in total. The Weizmann data set[¶] contains 28 persons' facial images with various angles, illuminations, and facial expressions. Excluding unclear images, we have chosen 52 images from each of 15 persons, resulting in 780 data items of 15 clusters. The size of each facial image is 88×128, resulting in a 11,264 dimensional vector. The InfoVisVAST data set[‖] is a document corpus of paper abstracts in IEEE Infovis (1995-2010) and VAST (2006-2010) conferences. It includes 515 documents encoded in 4,185 dimensions via a bag-of-words encoding after stemming and stop word removal.

4.2 Parallel Coordinates: Guiding beyond Two Leading Dimensions

When using dimension reduction in high-dimensional data visualization, the leading two dimensions of a dimension reduction method have been usually used to generate a single scatter plot while ignoring the other dimensions. Unlike these previous approaches, the testbed system first visualizes the reduced-dimensional data in the parallel coordinates view, and then two of these dimensions are interactively selected for the scatter plot view.

Although it is difficult to visually analyze data relationships in the parallel coordinates view, it can guide users in various ways. First of all, as shown in Figure 2(a), TSTG, e.g., LDA in this case, tends to separate different clusters in different dimensions. For instance, although the clusters '2' and '3' are not well separated in the scatter plot view with (1, 2)-dimensions selected, they seem to be separated from the other clusters in dimensions 7 and 4, respectively, based on the parallel coordinates view.

[§]http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits

[¶]http://www.wisdom.weizmann.ac.il/~vision/FaceBase

[‖]http://www.cc.gatech.edu/gvu/ii/jigsaw/datafiles.html

As another example, as shown in Figure 3, given the 10-dimensional results of PCA, the scatter plot view of (1, 2)-dimensions mixes up all the clusters together. However, hinted by the parallel coordinates view showing the peaks of the cluster '12' and '14' at dimensions 3 and 4, respectively, the scatter plot view of these dimensions turns out to give a well-clustered view. Such an observation is surprising because PCA is an unsupervised method, which does not take into account label information. This indicates that the leading two dimensions may not give enough information for high-dimensional data in visual analytics.

4.3 Effects of Alignment: Helping Comparisons between Visualizations

Trying various methods/settings on a given data set and comparing between different visualizations is in the heart of the testbed system, and the alignment functionality of the system supports this process. Figure 4 shows the effects of the alignment for clustering and dimension reduction.

In Figure 4(a), which shows the former, the three scatter plot views have identical coordinates of data items. With the different assignment of cluster labels, it is difficult to compare the cluster membership between the first and the third plots since the clusters have no correspondences in terms of the cluster colors and indices. After aligning the clustering, however, two different clustering results become much easier to compare between the first and the second view. For instance, compared to the original cluster labels shown in the first view, the original cluster '8' is shown to be merged to the original cluster '3.' The two subclusters of the original cluster '6,' which are shown in the bottom left and the top right in the first figure, are now split into two clusters in k-means clustering, and the former is shown to be merged to the clusters '4' and '10.'

On the other hand, Figure 4(b) shows the example of aligning dimension reduction. In this example, the cluster labels are unchanged for all data items in the three figures, but two different dimension reduction methods, TSTG and ISOMAP, are used. When comparing between the first and the third figures, which show the different coordinates generated by these two methods, it is difficult to recognize the correspondences between data items/clusters. Between the first and the second figures, whose dimension reduction results are aligned, one can perceive the correspondences in a much easier way. For example, the cluster '4' is shown to be close to the cluster '8' in TSTG, which is not the case in ISOMAP. Any data items in the cluster '6' are not located close to the cluster '7' in ISOMAP, but some data items between the two clusters overlap in TSTG. Such analyses cannot be easily made without the alignment.

4.4 Dimension Reduction: Supporting Multiple Perspectives

Different dimension reduction methods can reveal different aspects of data. To show an example, we now look into the first two figures in Figure 4(b) from the perspective of supervised vs. unsupervised methods. Given a certain assignment of cluster labels, a supervised method, TSTG, gives a clear overview in terms of cluster relationships since most of the clusters are shown relatively compact, as shown in the first figure. On the contrary, an unsupervised method, ISOMAP, may reveal different aspects of data. For example, the second figure indicates that the cluster '6' is composed of two distinct subclusters shown at the top left and the bottom right. However, when the data do not have a clear cluster structure, e.g., most of the text document corpora, unsupervised dimension reduction methods give the results similar to the second figure in Figure 3, which significantly reduces the utility of the scatter plot. In this case, supervised dimension reduction would be the only choice to start with in visual analytics.

Even with a single dimension reduction method with different parameter values, different aspects of data can be obtained. In Figure 5(a), one can see that the cluster '5' (the digit '4') of the Pendigits data set moves from the top left near the cluster '10' (the digit '9') towards the cluster '7' (the digit '6') as the ISOMAP parameter k increases. In general, ISOMAP with a smaller k value focuses more on preserving the local neighborhood relationships by making non-neighborhood distances longer. Based on the sample data of each digit shown in 5(b), it can be inferred that the digit '4' is represented much closer to the digit '9,' which forms their neighborhood relationships with small k values, than to the digit '6.' As the k value increases, the neighborhood relationship between the digits '4' and '6' starts to be formed, which is why they become closer at a bigger k value. In this way, varying the parameter values with the same method can further reveal different interesting insight about data.

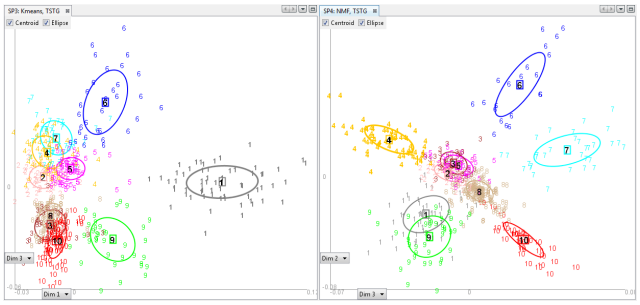


Figure 6: The scatter plot view of two different clustering, k-means and NMF, using TSTG for the InfoVisVAST data set. The right figure is aligned with respect to the left one for both clustering and dimension reduction.

4.5 Clustering: Combining Knowledge from Different Clustering

Clustering is a challenging task, and any single clustering method tends not to give fully satisfactory results. The testbed system can remedy this problem by enabling users to perform different clustering methods and obtain more meaningful clusters by comparing between them. Figure 6 shows the scatter plot views of TSTG with the cluster labels obtained by two different clustering methods, k-means and NMF. The InfoVisVAST data set are used, and the keyword summaries of clusters for each method are as follows:

k-means

1. graph, trees, node, layout, edge, draw, clusters
2. querying, interface, multiple, databases, expressive, temporal, magnification
3. document, text, collections, words, sequential, searches, information
4. multivariate, variable, data, aggregate, coordinates, multidimensional, flow
5. 3d, spatial, labelling, animation, map, coloring, display, information
6. treemaps, hierarchy, hierarchical, layout, focuscontext, spacefilling, algorithms
7. clusters, dimensions, image, visualization, measures, number, reduction
8. analytics, model, systems, video, decisions, information, framework
9. networks, traffic, arcs, diagram, social, internet, duplicate
10. collaboration, designed, histories, wikipedia, information, supports, story

NMF

1. graph, clusters, algorithms, methods, data, state, structured
2. querying, interface, databases, searches, temporal, multiple, data
3. document, text, image, content, information, collections, searches
4. dimensions, parallelize, coordinates, multivariate, multidimensional, datasets, scatterplots
5. 3d, spatial, landscapes, information, display, animation, spaces, encoded
6. treemaps, hierarchical, layout, ratio, algorithms, spacefilling, aspects
7. trees, hierarchy, node, genealogical, decisions, draw, layout
8. designed, model, information, analytics, framework, systems, data
9. networks, traffic, social, querying, analysis, data, flow
10. collaboration, analytics, wikipedia, analysts, supports, knowledge, shared

Among these clusters, the cluster '1' of the k-means clustering has a clear meaning of graph-related visualization, e.g., graph drawing, graph layout, and graph clustering. This cluster is also shown to be clearly separated from the other clusters in the left figure. As we perform brushing-and-linking on this cluster, it turns out that this cluster mainly corresponds partially to the clusters '1,' '6,' and '7' of the NMF clustering. Considering that these clusters contain the keywords, 'graph' and 'layout' and their separations from the other clusters in the right figure are not as clear as that of the cluster '1' in the left figure, one can regard the cluster '1' of k-means as a cluster with better quality.

On the other hand, in the NMF clustering, the cluster '4' seems to be clearly related to multi-variate/multi-dimensional data visualization. By brushing-and-linking on this cluster, we found it corresponds mostly to the clusters '4' and '7' of the k-means clustering, which makes sense based on their keyword summaries although they are relatively more ambiguous than the cluster '4' of the NMF clustering. This observation is also supported by their cluster separations in Figure 6 , which indicates a clearer separation of the cluster '4' in the right figure than that of the clusters '4' and '7' in the left figure.

As shown in these cases, one can apply different clustering methods and take full advantage of them by visually analyzing them in the testbed system.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the visual testbed system for dimension reduction and clustering in high-dimensional data visual analytics. The main contribution of our system is to bring a wide variety of traditional and state-of-the-art dimension reduction and clustering methods to visual analytics. The testbed system provides full control of these methods with interactive visual access to their results. In addition, our system offers a flexible extensibility for new data types and methods.

As future work, we plan to tackle a scalability issue. As the size of data gets bigger, their computational time takes even longer, which hinders real-time interactive visualizations. Another scalability problem is due to the limited amount of screen space. Even if the computational methods maintain efficiency, a large number of data items cause a clutter in visualization. These issues will be handled using various approaches, e.g., sampling, online learning algorithms, etc.

In addition, we plan to enhance the alignment capability by incorporating other advanced algorithms and user interfaces. To be specific, the currently used algorithms do not change anything in the reference view, and the Procrustes analysis does not change internal relationships within each visualization at all. This may limit the performance of alignment for easy comparison between visualizations when they are significantly different. To deal with this problem, we plan to utilize other advanced methods such as graph-embedding-based methods.[37]

ACKNOWLEDGMENTS

The work of these authors was supported in part by the National Science Foundation grant CCF-0808863. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Lowe, D., "Object recognition from local scale-invariant features," in [*Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*], **2**, 1150 –1157 vol.2 (1999).

[2] Keim, D., "Information visualization and visual data mining," *Visualization and Computer Graphics, IEEE Transactions on* **8**, 1 –8 (jan/mar 2002).

[3] Thomas, J. and Cook, K., [*Illuminating the path: The research and development agenda for visual analytics*], vol. 54, IEEE (2005).

[4] Jolliffe, I. T., [*Principal component analysis*], Springer (2002).

[5] Cox, T. F. and Cox, M. A. A., [*Multidimensional Scaling*], Chapman & Hall/CRC, London (2000).

[6] Fukunaga, K., [*Introduction to Statistical Pattern Recognition, second edition*], Academic Press, Boston (1990).

[7] Howland, P. and Park, H., "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 995 –1006 (aug. 2004).

[8] Kim, H., Drake, B., and Park, H., "Adaptive nonlinear discriminant analysis by regularized minimum squared errors," *Knowledge and Data Engineering, IEEE Transactions on* **18**, 603 – 612 (may 2006).

[9] Tenenbaum, J. B., Silva, V. d., and Langford, J. C., "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science* **290**(5500), 2319–2323 (2000).

[10] Roweis, S. T. and Saul, L. K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science* **290**(5500), 2323–2326 (2000).

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

[12] Park, C. H. and Park, H., "Nonlinear feature extraction based on centroids and kernel functions," *Pattern Recognition* **37**(4), 801 – 810 (2004). ¡ce:title¿Agent Based Computer Vision¡/ce:title¿.

[13] van der Maaten, L. and Hinton, G., "Visualizing data using t-SNE," *Journal of Machine Learning Research* **9**, 2579–2605 (2008).

[14] Ng, A., Jordan, M., and Weiss, Y., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems* **2**, 849–856 (2002).

[15] Kim, H. and Park, H., "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics* **23**(12), 1495–1502 (2007).
[16] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *Journal of Machine Learning Research* **3**, 993–1022 (March 2003).
[17] Wise, J. A., "The ecological approach to text visualization," *Journal of the American Society for Information Science* **50**(13), 1224–1233 (1999).
[18] Stasko, J., Görg, C., and Liu, Z., "Jigsaw: supporting investigative analysis through interactive visualization," *Information Visualization* **7**(2), 118–132 (2008).
[19] Cook, D. and Swayne, D., [*Interactive and Dynamic Graphics for Data Analysis: with R and GGobi*], Springer (2007).
[20] Jeong, D., Ziemkiewicz, C., Fisher, B., Ribarsky, W., and Chang, R., "iPCA: An Interactive System for PCA-based Visual Analytics ," *Computer Graphics Forum* **28**(3), 767–774 (2009).
[21] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., "The weka data mining software: an update," *SIGKDD Explor. Newsl.* **11**, 10–18 (November 2009).
[22] Asimov, D., "The grand tour," *SIAM Journal of Scientific and Statistical Computing* **6**(1), 128–143 (1985).
[23] Pisano, E., Zong, S., Hemminger, B., DeLuca, M., Johnston, R., Muller, K., Braeuning, M., and Pizer, S., "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms," *Journal of Digital Imaging* **11**(4), 193–200 (1998).
[24] Hastie, T., Tibshirani, R., and Friedman, J., [*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*], Springer (2001).
[25] Choo, J., Bohn, S., and Park, H., "Two-stage framework for visualization of clustered high dimensional data," in [*IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009.*], 67 –74 (oct. 2009).
[26] Wang, F., Sun, J., Li, T., and Anerousis, N., "Two heads better than one: Metric+active learning and its applications for it service classification," in [*Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*], 1022 –1027 (dec. 2009).
[27] Park, C. and Park, H., "Nonlinear Discriminant Analysis Using Kernel Functions and the Generalized Singular Value Decomposition," *SIAM Journal on Matrix Analysis and Applications* **27**(1), 87–102 (2005).
[28] Sammon, John W., J., "A nonlinear mapping for data structure analysis," *Computers, IEEE Transactions on* **C-18**, 401 – 409 (may. 1969).
[29] Zhang, Z. and Zha, H., "Principal manifolds and nonlinear dimension reduction via tangent space alignment," *SIAM Journal of Scientific Computing* **26**(1), 313–338 (2004).
[30] Weinberger, K. and Saul, L., "Unsupervised learning of image manifolds by semidefinite programming," *International Journal of Computer Vision* **70**, 77–90 (2006).
[31] Coifman, R. R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis* **21**(1), 5 – 30 (2006).
[32] Schlkopf, B., Smola, A., and Mller, K.-R., "Kernel principal component analysis," in [*Artificial Neural Networks - ICANN'97*], Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., eds., *Lecture Notes in Computer Science* **1327**, 583–588, Springer Berlin / Heidelberg (1997). 10.1007/BFb0020217.
[33] Lee, H., Kihm, J., Choo, J., Stasko, J., and Park, H., "iVisClustering: An interactive visual document clustering via topic modeling," *Computer Graphics Forum* **31**(3pt3), 1155–1164 (2012).
[34] Kuhn, H. W., "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly* **2**(1-2), 83–97 (1955).
[35] Hurley, J. R. and Cattell, R. B., "The Procrustes program: Producing direct rotation to test a hypothesized factor structure," *Behavioral Science* **7**(2), 258–262 (1962).
[36] Eldén, L. and Park, H., "A procrustes problem on the stiefel manifold," *Numerische Mathematik* **82**, 599–619 (1999).
[37] Choo, J., Bohn, S., Nakamura, G., White, A., and Park, H., "Heterogeneous data fusion via space alignment using nonmetric multidimensional scaling," in [*Proceedings of the 2012 SIAM International Conference on Data Mining (SDM12)*], 177–188 (2012).
[38] Asuncion, A. and Newman, D., "UCI machine learning repository." University of California, Irvine, School of Information and Computer Sciences (2007).