TripAdvisor^{N-D}: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail

Julia EunJu Nam and Klaus Mueller, *Senior Member*, *IEEE*

Abstract—Gaining a true appreciation of high-dimensional space remains difficult since all of the existing high-dimensional space exploration techniques serialize the space travel in some way. This is not so foreign to us since we, when traveling, also experience the world in a serial fashion. But we typically have access to a map to help with positioning, orientation, navigation, and trip planning. Here, we propose a multivariate data exploration tool that compares high-dimensional space navigation with a sightseeing trip. It decomposes this activity into five major tasks: 1) Identify the sights: use a map to identify the sights of interest and their location; 2) Plan the trip: connect the sights of interest along a specifyable path; 3) Go on the trip: travel along the route; 4) Hop off the bus: experience the location, look around, zoom into detail; and 5) Orient and localize: regain bearings in the map. We describe intuitive and interactive tools for all of these tasks, both global navigation within the map and local exploration of the data distributions. For the latter, we describe a polygonal touchpad interface which enables users to smoothly tilt the projection plane in high-dimensional space to produce multivariate scatterplots that best convey the data relationships under investigation. Motion parallax and illustrative motion trails aid in the perception of these transient patterns. We describe the use of our system within two applications: 1) the exploratory discovery of data configurations that best fit a personal preference in the presence of tradeoffs and 2) interactive cluster analysis via cluster sculpting in N-D.

Index Terms—High-dimensional data, coordinated and multiple views, zooming and navigation techniques, data transformation and representation, data clustering, visual analytics

1 Introduction

High-dimensional (high-D, N-D) data have become ubiquitous in a wide range of domains, such as science, finance, business, demographics, biology, and the like. Still, to date the interactive exploratory analysis of high-dimensional data remains a challenging undertaking, mostly due to the fact that high-D space is difficult to comprehend for humans. The following may serve as an explanation. It is widely thought [10] that human perception of the 3D physical world we live in is learned during infancy. During this time an unconscious inferential chain is established which is used to transform the input coming from the eye's optical system into the perception of 3D shape and relations. Nevertheless, this inferential process is by far not perfect and is based on many—not always fulfilled—assumptions. Numerous visual illusions readily demonstrate this. Since in our formative years as infants the concept of multivariate data spaces greater than 3D is typically not encountered or deemed relevant, no unconscious inferential reasoning chain for it is being learned and so we have now, as adults, a very hard time understanding and navigating a world of dimensionality greater than three.

It is often the navigation and orientation in high-D space that is most confusing to users, and likely for this reason most existing high-D data visualization systems perform their space navigation via the high-D version of the Manhattan distance, that is, only exchange one dimension at a time. This makes the exploration of multivariate relationships involving more than two dimensions difficult. Our system aims to overcome these shortcomings by providing a truly multivariate navigation interface in which users can transition across multiple dimensions at once intuitively.

Frequent tasks in data exploration are 1) the exploratory discovery of data configurations that best fit a personal preference in the presence of tradeoffs, and 2) a data partitioning (e.g., a clustering) that best fits an exploratory domain model. Our framework has been specifically developed to aid users in performing such operations directly in high-D space. It allows users to start from a key configuration, such as a projection or clustering, and then modify this standard configuration at their own will to locally optimize and fine-tune it. Here, a *key projection* can be obtained via any standard view optimization method such as projection pursuit or PCA, while a *key clustering* can be generated through the use of any standard clustering method, such as k-means, affinity propagation, and others.

Hence, the goal of our system is not to replace automated projection or clustering, but give users the ability to interactively refine the outcome of these automated routines or even tune their parameters so as to better fit their specific goals and expectations. As such, our system allows free-style

- *J.E. Nam is with the Microsoft Corporation, 15645 NE 92nd Way, Redmond, WA 98052. E-mail: namssi5@gmail.com.*
- *K. Mueller is with the Department of Computer Science, Stony Brook University (State University of New York), Stony Brook, NY 11794-4400. E-mail: mueller@cs.sunysb.edu.*

Manuscript received 22 Nov. 2010; revised 30 Aug. 2011; accepted 26 Jan. 2012; published online 17 Feb. 2012.
Recommended for acceptance by K.-L. Ma.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-11-0275.
Digital Object Identifier no. 10.1109/TVCG.2012.65.

out-of-the-box thinking but employs modern tools (i.e., cluster/projection algorithms) to do the rough work. Any results may then be captured into a formal model, such as SVM [17], HMM [9], or logic model [8].

Besides performing local operations, analysts also need to maintain a global overview of the high-D space, in terms of its highlights and features (called *landmarks*). Here, we are inspired by the emerging paradigm of *photo-tourism* [23] which uses maps to reference, index, and arrange large GPS-tagged photo collections, allowing for better management and sharing of these vast pools of data. Likewise, our framework also arranges the acquired visuals (the projections) of both key sites and user-discovered sites (called *snapshots*) into a spatially coherent reference frame. However, since in contrast to GPS-tagged photos our space spans a coordinate space greater than 2D and so does not allow for a direct 2D mapping, we define a projection similarity measure as a function of their orientation in hyper space and then use MDS for layout. This layout or map enables a better appreciation of the spatial relations among these high-D landmarks and snapshots, which can be subspaces, clusters, or a collection of optimal data points. Tours can then be built to narrate the findings to others.

Our system adheres well to Shneiderman's information-seeking mantra [20]: "Overview first, then detail on demand." Our extensible map of landmarks and snapshots provides the overview, while the detail (the landmarks/snapshots) can be interrogated with our local navigation tools. In the following, Section 2 discusses relations to existing work. Section 3 presents an overview while Sections 4 and 5 describe the map building and the local navigation frameworks, respectively. Finally, Section 6 illustrates the use of our system via two specific usage scenarios and Section 7 ends with conclusions and future work.

2 Relation to Existing Work

2.1 Visualization and Navigation

Various methods for comprehensive high-D data visualization have been proposed. The method of Parallel Coordinates [11] shows the entire space at once, but it serializes the dimensions, requiring an axis reordering to see alternative relationships in the variables. Space embeddings, such as Multidimensional Scaling (MDS) [15], also visualize the entire space, but their plots do not convey the data points in the context of their native attributes and thus all orientation hints are lost. Conversely, scatterplot matrices (SPLOM) [14] give an undistorted view, but they can only maintain two dimensions per tile and therefore cannot show multivariate relationships easily as these are distributed across the matrices. ScatterDice [5] gives users insight into 3D relationships by elegantly transitioning between two SPLOM tiles in a continuous fashion, which gives rise to a dynamic 3D point cloud projection display. Our local data explorer can be thought of as a generalization of this concept, enabling direct transitions in high-D space.

GGobi [4] uses projection pursuit [6] to generate interesting multivariate projections. Here, the dynamic transitions of the projection (hyper) plane allow users to experience high-D topologies as the "Grand Tour" [1] travels through hyperspace, but there is no clear notion of a *self-guided* tour. This is a significant shortcoming because "data explorers" want to actively control the exploration process. Our framework allows this self-initiated navigation control.

Most predominant are data axis-aligned 2D scatterplots. Since the number of such scatterplots grows in N^2, a number of authors have described methods to select more informative axis pairings given some criteria and only show these to the viewer—as opposed to show the entire SPLOM. The rank-by-feature system by Seo and Shneiderman [19] allows users to specify certain statistical criteria, such as correlation coefficient, scatterplot uniformity, and others, while Sips et al. [21] define a class consistency measure. The resulting *bivariate* scatterplots adhere to the common assumption that users will not understand scatterplots in which the data points are plotted as linear functions of more than two data variables, giving rise to *multivariate scatterplots*. Yet such projections, known as *biplots*, have been used in the statistics community for nearly 40 years [7] and are also available in the popular statistics package *R*. Typically, biplots are composed of a 2D coordinate system spanned by the two major principal component (PCA) vectors. The data samples are displayed as points and the variables are drawn as vectors or axes, expressed in terms of the PCA vectors or, as in GGobi, the basis vectors identified by projection pursuit. We will show that biplots indeed can lead to ambiguities and so deserve some caution. But we also show that these ambiguities can be resolved by allowing users, via an interactive navigation environment, to fine-tune the projections in light of the given task.

Overcoming projection ambiguities by ways of interaction has also been utilized in *star coordinates* [13]. Here, users can manually rotate and scale data axes to isolate data points or clusters that may originate from distant N-D locations but are ambiguously mapped to the same 2D area due to their similar dimension vector sums. These capabilities have been further advanced by Teoh and Ma [26] in the interactive construction of decision trees for high-D data classification. In our system the data axes do not require a direct manual manipulation, but change as a result of the user interacting with the multivariate navigation interface.

Finally, Yang et al. [28] construct maps of 2D axis-aligned projections as high-D space overviews, using the data correlations along the dimensions for the MDS layout of *dimension glyphs*. The glyphs are graphical representations generated by wrapping the 1D scatterplot along a dimension into a spiral. In contrast, our framework renders the glyphs from generalized 2D scatterplot projections and then lays them out via MDS using a correlation metric based on spatial similarity. This conveys the *spatial* relations among these projections, and so aims to render a better understanding of the high-D data space. Any local similarity in terms of the data relations is a secondary effect.

2.2 Dimension Management and Data Subspaces

A common remedy to manage the complexities of high-D space is to perform dimension reduction via PCA and the like and focus on the dimensions of most prominent variation. In [17] we used the three most dominant PCA vectors which resulted in 3D point clouds. We then exploited motion parallax to aid users in the perception of 3D relationships and enable them to perform visually assisted cluster analysis in

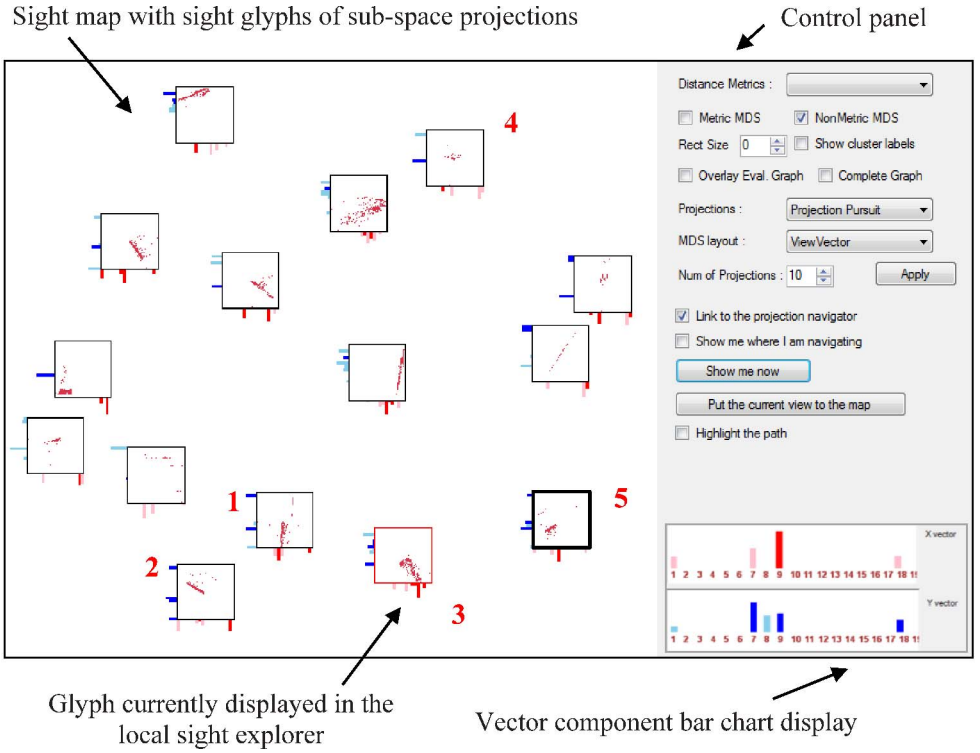


Fig. 1. Global sight map. On the left is the map with sight glyphs. The glyphs are distanced apart by a metric based on the similarity of the orthogonal N-D projection vector pairs generating the scatterplots in the glyph interior (see text for the similarity values for glyphs #1-4). Users can automatically generate interesting sights and insert new sights generated with the local sight explorer interface (see Fig. 3). Each glyph is decorated by bar charts showing the N-D coordinates of its projection vector pair (currently the user selected glyph #5 to have its vectors displayed in the vector component bar chart, control panel bottom).

3D—called *cluster sculpting*. We now generalize this work to allow for cluster-sculpting in high-D.

The iPCA framework of Jeong et al. [12] seeks to help users understand how the original data dimensions contribute to PCA space and also to the data clustering. Via a slider interface, users can interactively manipulate the contribution of each individual dimension and then observe the impact as transient changes in the scatterplot visualizations. In contrast, our framework enables users to perceive these contributions directly in the navigation interface.

However, once the number of dimensions grows large, global dimension reduction techniques are suboptimal. It is often better to discover the subset of the dimensions most relevant to a local clustering task, also called *subspace clustering* [18]. The ENCLUS framework by Cheng et al. [3] proposes entropy-based criteria to find interesting and minimal-dimensioned subspaces with high densities of data points. We make use of ENCLUS to extract the various subspaces from the high-D space, and then use our map as an intuitive framework to organize these subspaces and visualize their spatial relationships.

3 Overview

We demonstrate our system via two usage scenarios—a selection task and a clustering task—in conjunction with two data sets—a college ranking data set and an image segmentation data set. Both are discussed in Section 6. Our interface consists of two screens: 1) the global *sight map* which arranges both the landmark and the snapshot N-D projections (i.e., the sights) according to a spatial neighborhood metric, and 2) the local *sight explorer* which allows users to explore each of these projections via our high-D navigation interface. Any interesting views encountered there can then be inserted as snapshots into the map.

Global sight map (see Fig. 1). The sight map shows a number of projections each augmented with a set of colored bars arranged along the x- and y-axis. These bars indicate the relevance of the corresponding dimensions for this view, and they also allow users to quickly sense groupings and assess spatial similarities of neighboring views. Controls are available to 1) set the MDS layout metric, and 2) pick among different projection bases, such as PCA and projection pursuit, as well as clustering algorithms. The landscape map is linked with the local sight explorer. Users may 1) insert a new snapshot view acquired in the sight explorer, 2) specify an arbitrary tour passing through a set of mapped projections, and 3) use the map as an orientation aid by visualizing the current view in the explorer as a glyph that moves according to the map navigation path.

Local sight explorer (see Fig. 2). It consists of three major components allowing users to explore a given sight selected in the map: 1) scatterplot display, 2) a polygonal touchpad interface to control the scatterplot projections, and 3) information about the projection plane vectors. The polygonal touchpad enables users to smoothly tilt the projection plane in high-D space and so produce multivariate scatterplots that best convey the data relationships under investigation. Motion parallax and illustrative motion trails further aid in the perception of these transient patterns.

A video is available at [31] to show the system in action.

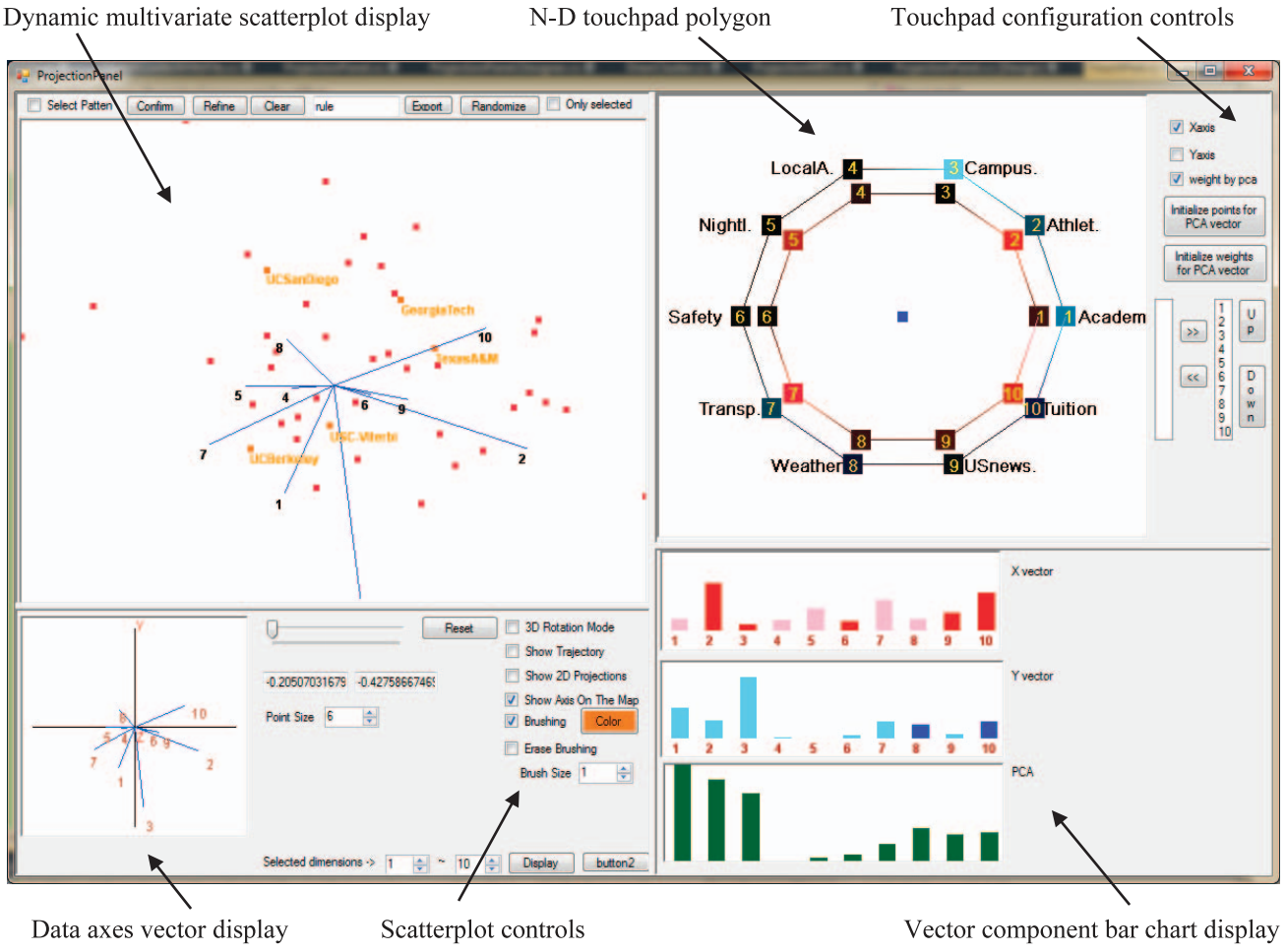


Fig. 2. Local sight explorer interface. The dynamic scatterplot display is controlled by the N-D touchpad polygon. Both are currently in a standard biplot configuration (using the two major PCA vectors as a projection basis). The PCA bar chart in the vector component display shows the magnitude of the 10 PCA vectors, and the other two bar charts show the components of the two major PCA vectors here selected for projection (these vectors can also be generated via projection pursuit techniques).

4 The Global Sight Map

After running ENCLUS to identify subspace clusters we employ projection pursuit (or PCA) to determine interesting scatterplot views in each of these subspaces. An *interesting view* is a 2D projection of the potentially high-D subspace that separates dense structures in the data well. These views can be general projections, i.e., they do not need to be aligned with the data axis vectors. The original projection pursuit [6] finds these views unsupervised via numerical optimization of a metric called *P-Index* which is the product of two parameterized measures: global spread and local density. Projection pursuit typically starts from the two major principal axes, two data axes, or two random orthogonal vectors. Some randomization in the numerical optimization and initial axis selection will yield a set of good views. We obtain one such set for each subspace.

We allow users to tune the P-Index parameters and so define the desirable characteristics of the views added into the global sight map (see Fig. 1). All sights on the map are characterized by one or more dense structures that are well separated and so already provide a good appreciation of the subspace clusters and their shapes. Touring the sight with the local sight explorer will then enable more comprehensive insight into the high-D structure of the clusters.

Map construction. To construct the map, given M views, we first compute the $M \times M$ distance matrix for all pairs of projection views and then determine their positions in the sight map via MDS. For this we need a metric to determine these pairwise distances. Each view S_i, $1 \leq i \leq M$, has two orthogonal N-D axis vectors which we call *projection plane axis (PPA) vectors*, PPA_x and PPA_y. In case of subspace decomposition, dimensions that are not contained in a subspace will have a zero value in the N-D vector. To characterize each view by a single vector S_i we concatenate PPA_x and PPA_y into a vector of length $2N$. Then, for two projection views S_1 and S_2 we compute their similarity as the Euclidian distance of S_1 and S_2. Note that view rotations and also axis reflections will be rated as dissimilar in this scheme. This is intended because these dissimilarities show semantically different relationships, e.g., sorting college data by low versus high tuition, or can be part of a tour. If this is not desired, one may take the absolute values of the PPA vector components before computing the distance matrix.

Sight glyph design. The sights are abstracted into glyphs, which are constructed by their scatterplot projections and augmented by N-D space location information.

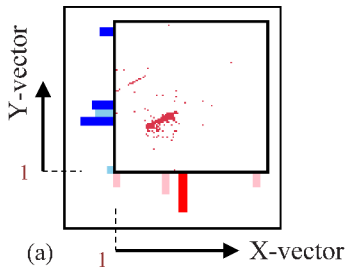


Fig. 3. Sight glyph (#5 in Fig. 1) showing the projection scatterplot and along its sides the relevance of the data dimensions in the x and y PPA vectors that generate the scatterplot. The corresponding vector component bar char display is shown in Fig. 1, control panel (bottom).

The latter is not required for conventional 2D sightseeing maps, but it is needed here since the nonlinear MDS embedding does not allow for a meaningful placement of orientation "beacons" in the map, at least not without further distortion. Instead, we choose to make the sights themselves our orientation beacons. We place bar charts on the left vertical edge and on the bottom horizontal edge, one bar for each dimension. In this chart, the x-axis and y-axis bars encode the components of the PPA_x and the PPA_y vectors, respectively. They indicate the significance of a data attribute for this projection axis. Both bar charts have their origins placed at the lower left corner of the sight glyph, which is similar to a standard x-y plot. We use a faint color to represent negative axis directions, reserving bar height to encode magnitude. Fig. 3 shows the augmented glyph for a given projection. A more detailed bar chart with axis labels appears in the local sight explorer (Fig. 2) and optionally also in the global sight explorer interface (Fig. 1) where it gets updated upon selecting or hovering over a sight glyph. The simple bar charts provide for a compact glyph representation that is both intuitive and comparable across sights. It also scales well to a large number of sights. On the other hand, the single annotated chart in the control panel gives more detailed information when needed. Users can also query dimension names and values by hovering over the corresponding bar location in the sight glyphs.

Map interaction. The sight map forms the "tour map" and the local sight explorer is available to examine each sight or "destination" in detail. Clicking on one of these destinations activates the current view, denoted by coloring its frame red, and the scatterplot view is loaded into the local sight explorer where it can be manipulated. Users can examine any view in the tour map in this way, but they may also use the distance and orientation information to connect the sites in some order. In this way, the map allows "tour designers" to plan an exploration tour for "customers." So, unlike when traveling with the Grand Tour, "tourists" now have a map by which they can compare the location of the sights and predict the time for travel. The tour can be accelerated by simply clicking on the next icon along the tour. These steps often are revisited after gaining insights into a certain destination.

Discussion. In Fig. 1, we have taken the absolute values of the PPA vector components such that rotation or reflection about a data axis will not affect the similarity measure. Thus, only magnitude determines the layout in this example. We observe that the bar chart augmentation conveys much information on the differences in N-D projection orientation, both on a local and on a global scale. We also see that views with similar N-D space orientation—indicated by similar axis bar chart configurations—indeed cluster in close neighborhoods. This is further confirmed by comparing some numerical similarity distances. In Fig. 1, view 2 and 3 (see annotations in red) have roughly the same similarity score and they also have similar map distances with respect to view 1 (0.68 versus 0.70), while view 1 and view 4 have a much larger score (2.68—higher values map to lower similarity). It is interesting to see that the scatterplots of even nearby views are often quite different. This is due to the quick transitions in the projections of high-D structures which can be readily experienced in the local sight explorer's dynamic scatterplots.

5 The Local Sight Explorer

The local sight explorer interface, shown in Fig. 2, consists of three dynamic visualization panels (the *multivariate scatterplot display*, the *data axis vector display*, and the *vector component bar chart display*), one interaction interface (the *touchpad polygon*), and two control panels (one each for the scatterplot and the touchpad). More specifically:

The dynamic multivariate scatterplot display (described in Section 5.2) projects—in biplot style—both the data and the axes of the active dimensions into the PPA-vectors, which form the x- and y-axes of the display.

The data axis vector display visualizes the projections and labels of the active dimension axis vectors in isolation.

The vector component bar chart display features three bar charts. The bottom chart shows the PCA spectrum where each bar represents the magnitude of the corresponding PCA vector. The PCA spectrum conveys the extent of a cluster in terms of the data dimensions. Users can select any two of these vectors, or vectors obtained by projection pursuit, as initial PPA vectors. For PCA, typically one would use the two most significant PCA vectors, but often there might be more than two major vectors, as is the case here. The other two bar charts show the x (red) and y (blue) components of the current PPA vector pair, as set via the touchpad. A faint color indicates a negative direction.

The N-D touchpad polygon (described in Section 5.1) is here configured with 10 vertices, one for each active dimension. It is used to control the orientation of the two PPA vectors by simply translating the two corresponding pointers in the polygon's interior. In this current initial configuration these vectors are the two major data PCA vectors. This is a standard biplot setting and both pointers locate in the polygon center—note that the blue x-axis vertex occludes the coinciding red y-axis vertex. The inner and outer rings of the polygon are due to the vectors representing the x-axis and the y-axis of the scatterplot, respectively. The shading of the vertices indicates the weight of the dimension axes, which in this case corresponds to the PCA vector components. The weighting ensures that the local space explorations enabled by moving the vertices stay reasonably close to the distribution spread of the selected PCA (or projection pursuit) vectors—selecting another basis will yield a different view orientation neighborhood.

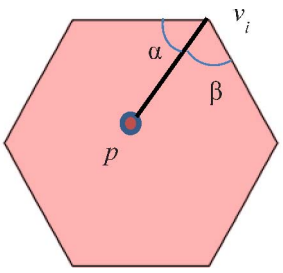


Fig. 4. N-D touchpad interface. Using generalized barycentric coordinates, the interpolation weight of dimension axis vertex v_i for a PPA pointer p is related to angles α and β, and is inversely related to the distance $|p - v_i|$.

The two control panels are used to configure the touchpad polygon and the scatterplot display, respectively.

5.1 The N-D Touchpad Polygon

5.1.1 Basic Theory and Operation

The N-D touchpad interface (see Fig. 4) builds on a paradigm we proposed in 2008 [8]. A similar interface was also proposed shortly after by Talbot et al. [25]—they call it *Linear Combination Widget*—designed for building ensemble classifiers in machine learning. Our interface extends the familiar 2D interaction mapping on mouse touchpads often found on laptops to support higher dimensions. As mentioned above, the N-D touchpad consists of two N-sided equilateral polygons—one each for the PPA_x and PPA_y vectors. Using geometrical arguments, the method of generalized barycentric coordinates of Meyer et al. [16] can be used to interpolate a value in the polygon interior from the values assigned to its vertices. Referring back to Fig. 4, the interpolation weight w_i of vertex v_i for an interior point p is

$$w_i = \frac{\cot(\alpha) + \cot(\beta)}{\|p - v_i\|^2}. \quad (1)$$

Given N vertices the interpolated value pv at p is then

$$pv = \sum_{i=1}^{N} v_i a_i \text{ where } a_i = \frac{w_i}{\sum_{k=1}^{N} w_k} \text{ and } \sum_{i=1}^{N} a_i = 1. \quad (2)$$

This mechanism allows one not only to interpolate spatial coordinates in the native 2D polygon domain, but also to interpolate vectors of higher dimensions by assigning these vectors to the vertices (see, e.g., [22]). More formally, with N being the number of dimensions, we can write this expression in matrix form as $P = VA$, where V is an $N \times M$ matrix of M column vectors v_i, A is the M-long column vector with coefficients a_i, and P is the N-long column vector that is being interpolated. Typically $M = N$ and each vertex has one dimension unit vector assigned to it.

The touchpad then enables a user to vary the PPA_x and PPA_y vectors, and consequently the multivariate projections, by transitioning a pointer (one each for PPA_x and PPA_y) within the polygon. By moving the pointer closer to a certain dimension, the dimension's weight will grow larger in the vector, effectively "attracting" that dimension axis to this PPA-vector. Moving the x-axis and y-axis pointers toward different dimension vertices (or combinations of these) will visualize their correlations in the scatterplot.

When mapping a view's PPA-vectors to the touchpad, we first multiply each dimension vector by its dot product with the PPA vectors and use the resulting vectors in V. Then, we compute $A = V^{-1}P$ for both x and y pointers. While this will not always yield coordinates consistent with the 2D spatial constraints of (1), automated reordering of the vertices via quick trial and error can often help. This procedure places the pointers in a general position within the touchpad polygon. All of the examples shown in Section 6 have successfully used this mapping.

The interpolation function defined by the method of generalized barycentric coordinates is smooth, which in turn guarantees smooth transitions in the dynamic scatterplot display. However, the geometric weight equation (1) is only valid for movements strictly inside the polygon. Any point contained on or very close (within a small margin ε—we use 0.05) to the polygon boundary will give rise to numerical instability. In these cases we fall back to linear interpolation along the corresponding edge. This has been suggested by Meyer et al. and is desirable for our purposes because in this case we wish to enable the user to only vary the scatterplot as a function of the two edge variables.

5.1.2 Extended Theory

Although the above mapping has worked well in our initial work [8], the touchpad as described above cannot specify all possible PPA vector orientations in a high-D data space once the number of dimensions grows beyond 3. This is also true to some extent for the widget proposed by Talbot et al. Per our definition, a PPA vector is a unit vector in a data space that is bounded by a hyper-box normalized to $[\pm 1]$—one can always scale the result back during display. The PPA vector orientation is defined by the coordinates on a hyper sphere of $M \leq N$ dimensions and radius 1, where N and M are the dimensionality of the data and the hyper sphere, respectively. The hyper sphere is uniquely specified by the set of M orthogonal data axis vectors, and the more axis vectors are included, the greater the dimension range of the PPA vector. The number of unique hyper spheres is given by $N!/(M!(N\text{-}M)!)$. Thus, a hyper sphere with $M = 1$ dimension can only specify a single PPA vector orientation—the one along the defining data axis. Table 1 lists all such configurations for the 4D case ($N = 4$): the dimensionality M of the hyper sphere, the number of unique hyper spheres for each M, and the associated dimension sets. For $M = 1$ there are four 1D hyper spheres, for $M = 2$ the hyper sphere is a 2D disk and there are six unique axis-aligned disks, and for $M = 3$ there are three unique 3D spheres.

One might ask, why not just choose $M = N$ and use an N-sided polygon as the touchpad. This is an inferior choice since such a general polygon will not allow the specification of all possible PPA-axis orientations. For example, assuming an ordered (data axis $\rightarrow$ polygon vertex) configuration (see Fig. 5a), one could not specify a PPA vector that is a linear combination of data axes 1, 2, and 4 without including effects of data axis 3 as well. In fact, any polygon with $M > 3$ will give rise to uncovered orientation angles. Furthermore, the order of the vertices also matters when $M > 3$. As Table 1 indicates, for $M = 4$ there are three different vertex orderings

TABLE 1
The Set of Hyper Spheres with Dimensionality 1-4 for a 4D Space and Their Touchpad Navigation and Setup Schemes

dimensionality	1	2	3	4
# hyper spheres	4	6	4	1
hyper sphere dimension sets	1,2,3,4	12,13,14,23,24,34	123,124,134,234	1234, 1342, 1432
touchpad navigation	move to vertex	move along edge	move within triangle	move within poly, re-order

each giving access to different, but partially overlapping regions of the 4D space. To illustrate, consider the interior point p in Fig. 5a. Switching vertex 1 with vertex 4 to change the vertex order from 1234 to 4231 will achieve weight coefficients, and therefore PPA vector orientations, not possible with the original vertex ordering. In general, there are $M!/(2 \cdot M)$ such orderings. Here, we divide by M and 2 since the vertex order is circular and due to this, the direction of the ordering is also irrelevant.

5.1.3 Full Operation

Only the complete set of all possible unique vertex configurations can ensure that all PPA vector orientations in the N-D hyper sphere can be covered. The touchpad interactions associated with these (for $N = 4$) are listed in the bottom row of Table 1. Setting the vertex weights to negative numbers enables all hyper sphere segments to be reached.

Users can select the hyper sphere currently navigated by assigning the corresponding dimensions to the touchpad vertices. Picking these dimensions from the scrollable attribute list is done using the '≪' and '≫' buttons in the touchpad configuration controls (see Fig. 2).

Note that the configuration of Fig. 5a treats all vertices equally when navigating the space. However, if one wishes to control the influence of, say, dimension 3 more subtly, reaching into a subspace that is only mildly influenced by dimension 3, then one could either move vertex 3 farther out (Fig. 5b) which would yield a nonequilateral polygon or add a small weight to vertex 3 (Fig. 5c) and use this weight in the generalized barycentric interpolation. We have chosen to use the latter option (Fig. 5c) in our implementation to ensure a fixed geometry of the touchpad polygon. This weight can be user controlled via a simple slider at each vertex. Equation (2) can then be rewritten as follows:

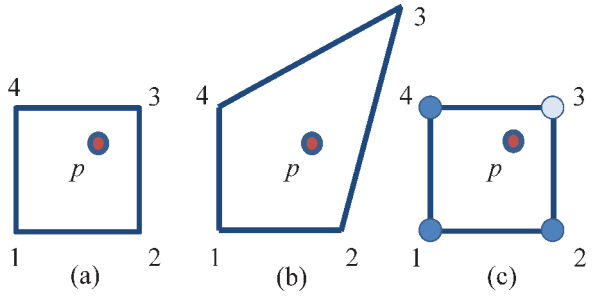


Fig. 5. Influence of dimension 3 on the interpolation of point p: (a) standard influence, (b) reduced influence by stretching vertex 3 far out, (c) reduced influence by assigning a small weight to vertex 3.

$$pv = \sum_{i=1}^{N} v_i c_i a_i \text{ where } a_i = \frac{w_i}{\sum_{k=1}^{N} w_k} \text{ and } c_i = \frac{b_i}{\sum_{k=1}^{N} b_k} 1. \quad (3)$$

Here, the b_i are the additional weights at the vertices, which are subsequently normalized to yield the vertex coefficients c_i. We initialize these weights using the two selected PCA vectors. More specifically, for PPA_y we use the first such vector, while for PPA_x we use the second one. We then compute the dot product for each dimension vector with the respective PCA vector to obtain the weights. A similar procedure applies if the PPA_y and PPA_x vectors are instead obtained via projection pursuit.

This weighting places the PPA pointers into the touchpad center when they are assigned—during initialization—to the PCA or projection pursuit vectors. By giving higher weights to more relevant dimensions, the space navigation tends to stay better on target with respect to the spread of the interrogated point distribution. This is demonstrated in Fig. 6 where we show, for a fixed PPA pointer position, the corresponding scatterplots for both the unweighted (Fig. 6a) and the weighted case (Fig. 6b). The latter shows the distributions much more clearly and well spread out.

A reordering of the vertices can be accomplished by selecting a dimension on the list and then using the "up" and "down" buttons in the touchpad configuration controls (Fig. 2). A good vertex ordering is one in which more highly correlated dimensions are adjacent to one another in the touchpad. This will enable navigation to the subspace in which the correlated structure is well expressed. Of course, there can be multiple such structures and with conflicting orderings. We have recently developed a framework [30] for dimension ordering in parallel coordinates that lays out dimensions by their correlations into a 2D map and then allows users to configure routes through this dimension space. A traveling salesman solver assists by configuring the shortest route—the one in which the sum of correlations is maximized—taking into account any preferred sub routes. Work is underway to tie this with the touchpad.

Finally, although users are permitted to freely move about the space choosing the PPA-vectors via the touch pad interface, one must still enforce that the two PPA-vectors remain mutually orthogonal. To ensure this, for example, when moving the PPA_y pointer, we project this vector onto a vector $PPA_{y'}$ that is closest to a vector orthogonal to the current PPA_x pointer (or vice versa)

$$PPA_{y'} = \frac{y'}{|y'|} \quad y' = PPA_y - (PPA_x \cdot PPA_y) PPA_x. \quad (4)$$

Fig. 9 shows the adjusted pointer as a dark blue dot (near the unadjusted aqua dot).

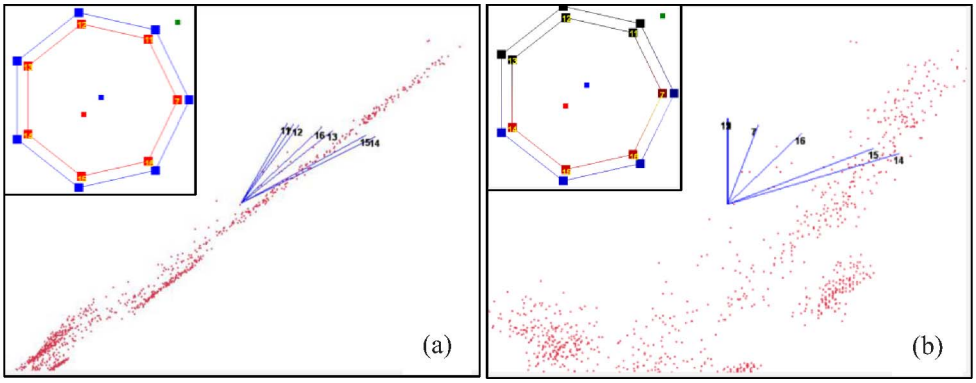


Fig. 6. Effect of using distribution-sensitive weights for the dimension axes (the inserts show the corresponding touchpad polygons): (a) no weights can result in poor projection quality once the PPA pointers are moved away from the initialized views, (b) weighting ensures that the space exploration maintains good spatial distribution patterns.

5.2 The Dynamic Multivariate Scatterplot

5.2.1 Basic Operation

This plot projects both the data and the axes of the active dimensions into the PPA_x and PPA_y vectors which are aligned with the x and y direction of the plot, respectively. The projection is done by a simple dot product. Apart from data correlations, the plot also shows which dimensions are not or only mildly expressed in the data relationships. They project as very short axes in the axis plot. The touchpad is used to locally tilt and transform the PPA_x and PPA_y vectors in such a way that individual dimension axes can be pulled apart, aligned, or their influence increased or reduced, such that correlations, tradeoffs, and preferred configurations in the data can be revealed.

In [8] we showed that the dynamics of motion can be very useful in revealing interesting high-dimensional patterns. This is akin to the perceptual depth cues in 3D viewing situations where motion parallax has been shown to provide stronger depth cues than stereo vision—at least for scene objects not in the very-near range [27]. Motion parallax has also been exploited in the projection pursuit of GGobi's Grand Tour. However, in the Grand Tour users are mostly confined to watching motions until interesting projections appear. While there is some level of interactive navigation control, users cannot change the projection plane orientations in arbitrary ways. Our N-D touchpad, on the other hand, provides significantly more freedom in that respect. The direct navigation that it affords allows users to easily transition to these interesting projections and at the same time it also gives them a sense of location and orientation, which is important for navigation tasks.

5.2.2 Extensions

We have extended the basic framework described in Section 5.2.1 in three important ways. As an illustrative example, we use a simple dynamic exploration of a network traffic data set. This data set was obtained from the MAWI Working Group Traffic Archive [33]. It represents traffic data captured over an hour, in tcpdump format. The data set contains 15 dimensions and we have chosen source IP, destination IP, and time stamp for the exploration.

Color-label enabled N-D tracking. In our experiments we found that color labeling can provide an important tracking clue for users. So, before transitioning the projection plane, we divide the current projection into grid cells and assign a unique color to each. In practice we randomize colors such that no identical color is assigned to neighboring cells. We assign this same color to all N-D points projected into this cell, no matter if overplotting has rendered them occluded. Upon transitioning, this color will then identify patterns evolving from previously coinciding points. Fig. 7 shows three screen captures of this dynamic exploration while the user fixes the PPA_x vector onto the source IP dimension and moves the PPA_y vector from destination IP to time stamp. Fig 7a shows the random colorization of the data points projected as a function of source IP versus destination IP only. When we then move the y-axis point from the destination IP dimension to the time stamp dimension we can easily observe the N-D trajectories of the exposed data points by their identical color labeling (Figs. 7 and 7c). The data points labeled with the same color tell us that there were packet exchanges between the same source IP and destination IP over some period of time. Some are short and some are longer, as can be easily compared by the length of the colored streak line.

Motion trail enabled N-D tracking. The color labeling is most effective for point distributions that are somewhat striated in N-D space (as is the case for the network data set). For more generalized data topologies, we simulate the streaks by adding a motion trail to the color labeling. We find that this conveys the trajectory of the points during the projection plane transitioning very well. The motion trail is rendered as a thin equicolored tail attached to the current point position. This effect is provided in both the scatterplot and the touchpad. We informally find that the dynamic exploration with the motion trail reveals trajectory patterns that are difficult to identify otherwise, as is shown in Figs. 8 and 11. The motion trail frees the user from having to transition back and forth repeatedly.

6 Application Examples

As mentioned, $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ informally relates high-D space exploration to the activities undertaken before and during a sightseeing trip. It decomposes this procedure into five major tasks:

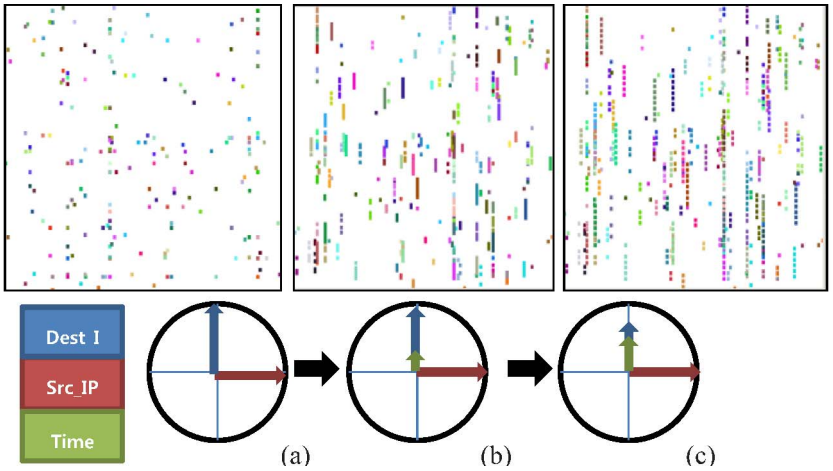


Fig. 7. Color-label enabled N-D tracking using the network traffic data set. (a) Initial scatterplot of dest_IP versus src_IP—all packets projecting onto similar (x, y) coordinates are assigned the same color, (b) tilting the projection plane to reveal time relationships—formerly superimposed packets now visualize as shorter or longer streaks along the y-axis, (c) tilting the plane further reveals individual packets.

1. Pick the sights: use the map to identify sights and landmarks of interest along with their location
2. Plan the trip: connect these landmarks and sights of interest along a specifiable path.
3. Go on the tour: travel along the route.
4. Hop off the bus: experience the sights, look around, zoom into detail, take snapshots of interesting personal observations.
5. Orientation and localization: regain bearings using landmarks, correct for mapping errors.

We demonstrate our framework via two different task scenarios: 1) the exploratory discovery of data configurations that best fit a personal preference in the presence of tradeoffs, and 2) the interactive cluster analysis via cluster sculpting in N-D, refining clustering results obtained with unsupervised methods to fit personal goals. The two application examples that we have chosen to demonstrate our framework were selected for their intuitive themes. The first example uses mainly the local sight explorer, while the second makes use of the complete framework.

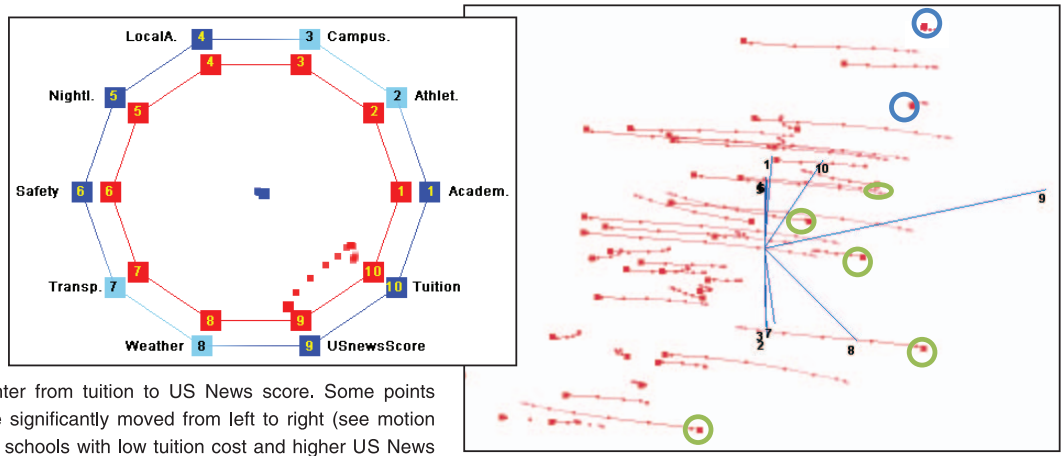
6.1 Discovering Preferred Data Configurations

A good representative example for a selection task involving a large and diverse set of competing factors is that of selecting a college or university to study at. Here, one must balance among many different goals, such as academic standing of the schools, tuition, academic, and social environment, the cost of housing and travel, reputation, tradition, and so on. This is inherently a multivariate data exploration problem. Tradeoffs will certainly have to be made, but in this process one would desire to have all facts on the table at all times to preserve a full mental picture. The data set we use is an amalgamation of data obtained from two different sources: the College Prowler website [34] and US News & World Report [35]. The former ranks each school across the 20 most relevant campus life topics. We took the top 50 colleges from US News and eight topics from College Prowler, including academics, athletics, campus housing, local atmosphere, nightlife, safety, transportation, and weather. These attributes are ranked from A+ to D− and consequently we mapped the range (A+ to D−) to values (1.0 to 0.0). We also added two further dimensions—2009 US News score and tuition. Both are also normalized to (1.0 to 0.0). We shall denote a dimension as X_i.

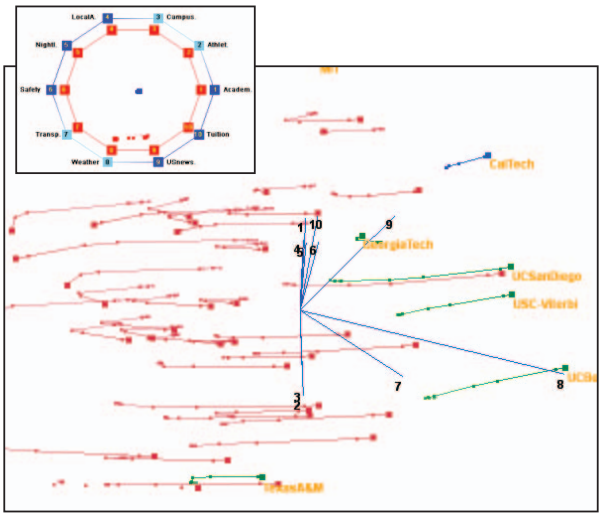
The shortcomings of biplots. The sight explorer in Fig. 2 shows a standard biplot of the data, generated by projecting all data points and dimension axes into a basis spanned by the two major PCA vectors. We only label a select number of schools to minimize text clutter, and Table 2 lists the scores of these schools for the 10 attributes we chose. As prescribed, biplots allow one to appreciate multivariate relationships in the data. However, we can make a number of interesting observations. For example, although USC-Viterbi has the highest tuition (X_{10}) of the five schools it plots near the coordinate system origin, while Texas A&M which has a low tuition plots in the higher region along this dimension. This due to the fact that USC-Viterbi also has a high transportation score (X_7) which points the other way in the biplot and so "pulls" the school to the center. Conversely, Texas A&M has a low transportation score and so is "pushed" into the high-tuition direction. In contrast, Georgia Tech, places well in these respects.

Can we make biplots more robust? While it goes undisputed that biplots are a powerful mechanism to convey multivariate relationships, their static layout that is pinned to the PCA vectors gives users no means to resolve and become aware of the fallacies they potentially produce. Our touchpad navigation interface delivers just that—an instrument by which biplots become interactive and tunable, allowing users to explore and verify the plotted relationships and also personalize the multivariate projections to fit their own objectives. We shall illustrate this process next, with the best college search as a usage scenario.

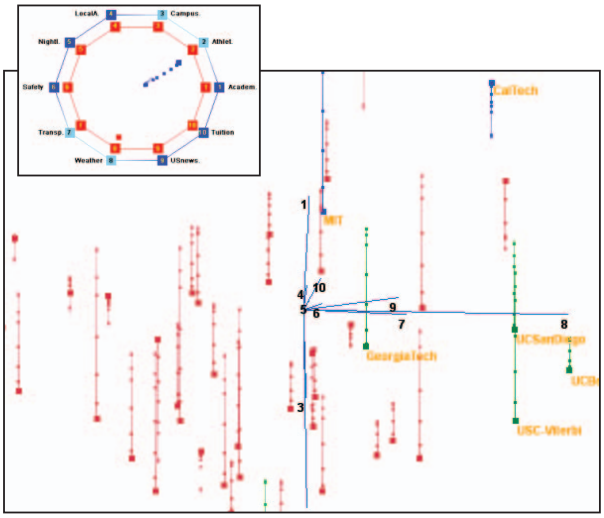
Usage scenario. Let us follow Ben, a high school counselor with visualization experience through the process of selecting a well-suited college for Tim, an honors student at the high school. First, to gain an overview of the data, Ben prepares a standard sight map using projections onto PCA vectors (see Fig. 2). Based on this initial map, he now decides to explore the data set further with Tim, augmenting the map with additional snapshots.



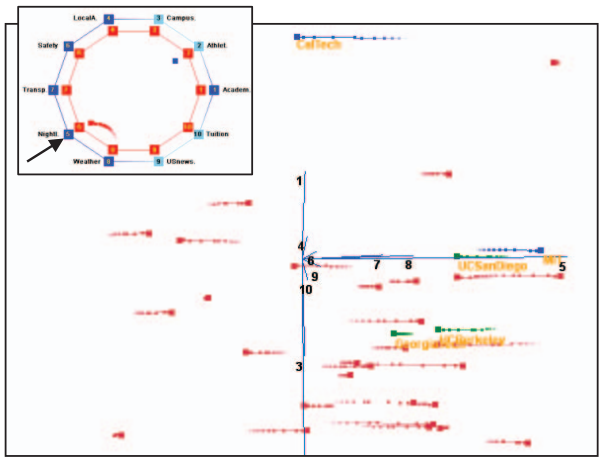
(a) Move PPA_x pointer from tuition to US News score. Some points (circled green) have significantly moved from left to right (see motion trail). These are the schools with low tuition cost and higher US News score. Most data points move into the opposite direction: they have high tuition but low US News score. The points circled blue stay put: they have both high tuition and high US news score. We brush the first group in green color to track them further.



(b) Move PPA_x pointer from US News to weather score. Via the motion trail, the original position for the US News score can also be appreciated. All UC colleges and USC have exceptionally great weather scores while Texas A&M's weather score cannot make up for its lower US News score. Georgia Tech's weather is about the same than its US News score so it does not move much.



(c) Move PPA_y pointer from center to the athletics score (the student is a sports fan). Note that we have moved the y-axis and the athletic axis points down which means data points lower down have a higher score on this dimension. While the UC colleges all have great athletic scores, Caltech has a very low score here. So from this combined point of view (weather/athletics) USC Viterbi looks best followed by UC Berkeley and Georgia Tech. The candidates seem to emerge now.



(d) Reordering dimensions such that nightlife places next to weather (see arrow) and moving PPA_x pointer toward the nightlife score. Both the UC schools and Georgia Tech have moved from right to left revealing good weather but relatively low nightlife scores. Indicated by the motion trail, Georgia Tech's movement is shorter than UC Berkeley's suggesting that its two scores are more balanced. Both schools end at similar x-locations suggesting a larger trade-off for Berkeley between nightlife/weather. A trade-off can also be made between UC San Diego and UC Berkeley. UC San Diego has a lower athletics score than UC Berkeley (y-axis position) but its scores are higher for both weather and nightlife (see trajectory).

Fig. 8: Best College selection example. (a)-(d) show four subsequent interactions in the touchpad along with the generated dynamic scatterplot, See caption for a detailed description of each.

Fig. 8. Best College selection example. (a)-(d) show four subsequent interactions in the touchpad along with the generated dynamic scatterplot, See caption for a detailed description of each.

TABLE 2
Attribute Values for a Subset of Five Schools from the College Data Set

School	Academics	Athletics	Housing	Atmosph.	Nightlife	Safety	Transport.	Weather	USNews	Tuition
UC Berkeley	10	9	8	9	8	7	10	11	89	14,998
Georgia Tech	10	11	5	10	9	7	7	8	86	22,188
USC Viterbi	10	2	8	11	12	7	8	11	77	22,734
UC San Diego	9	8	6	11	9	11	6	12	72	14,694
Texas A&M	7	11	8	7	7	9	5	8	69	8,712

Fig. 8 demonstrates Ben's exploration path in his mission. He learns from Tim that at the moment his overall preferences are low tuition and a good US News score, but that he has no clear preference beyond these attributes and has an open mind. Fig. 8a shows Ben transitioning the PPA_x vector from the tuition to the US News score dimension—all other dimensions have small weights on the PPA_y vector and so have little influence. As visualized by the motion trail trajectory, some data points (green circled) move significantly from left to right while other data points move into the opposite direction. In fact, these two groups are crossing during the transition. These (green circled) schools are the ones to watch. They have good US news scores but also relatively low tuition, compared to other well-ranked schools (blue circled) that did not move much since they have both high tuition and high US News score. Having identified these "nuggets" Ben is now ready to consider other factors, assisted by Tim. He first brushes these (green circled) schools to better follow their paths.

Next, Figs. 8b, 8c, and 8d show a set of explorations (see captions) in which Ben and Tim identify a set of viable schools and, using the motion trails, also learn about tradeoffs each school has with regards to the various factors. Large motion blur indicates high tradeoff because the score changes rapidly when transitioning from one factor to the other.

The explorations leave the touchpad vertices in an ordering that already reveals certain factor groups in which tradeoffs appear tolerable. In the setup shown in Fig. 9 there are three factor groups: 1) weather, US News score, and to a lesser extent nightlife, 2) athletics, academics, and to a lesser extent campus life, and 3) tuition. This requires a 7-factor visualization that respects these tradeoffs. It can be achieved by moving the PPA_x and PPA_y pointers away from the boundary (Fig. 8d) and into the polygon (Fig. 9) at the locations that quantify the degree of acceptable tradeoff. Ben also transitions back and forth with the touchpad to get a sense for the biplot mapping error.

The significance of the variables is also apparent in the scatterplot where the length of an axis indicates the amount of impact this attribute has in the view. For example, the weather (X_8) and US News score (X_9) dimensions are aligned with the x-axis direction at similar length, but nightlife (X_5) is less significant and so has a reduced length. Local atmosphere (X_4) has a very small length, meaning that it is not considered an important criterion here.

This multivariate scatterplot is essentially the end product of the exploration session. It tells Ben that USC Viterbi has the best score all together but that it also has the highest tuition. He points out to Tim that Georgia Tech and UC Berkeley also have relatively good scores for all factors significant to him, but at lower tuition costs than USC. Tim now decides that he really values athletics and academics more than weather and US News score, and so he picks Georgia Tech. We recall that this school was also quite stable in the transitions, and so seems well factor-balanced.

Discussion. Transitions enabled by moving across a traditional scatterplot matrix in a row/column-wise fashion, as elegantly implemented in ScatterDice, can only change

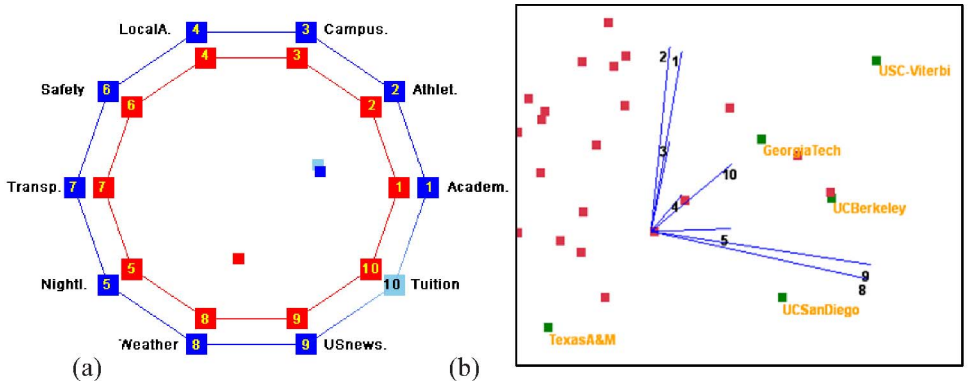


Fig. 9. Final multivariate scatterplot for the Best College selection example, considering seven (significant in this configuration) factors at the same time. Panel (a) shows the touchpad, panel (b) shows the scatterplot. Based on the prior excursions we create two 3-factor groups in which tradeoffs are tolerated: (weather, US News score, and with less importance nightlife) and (athletics, academics, and with less importance campus life). Tuition plays a special (but less weighted) role and is placed as a seventh isolated factor between these two. We see that given the two first factor groups, USC-Viterbi seems to be the best college but it has the most expensive tuition. Georgia Tech (despite somewhat higher tuition cost) and UC Berkeley also appear to be good alternatives. Given all these options, the student decides that he actually values athletics and academics higher than weather and US News score and so he picks Georgia Tech.

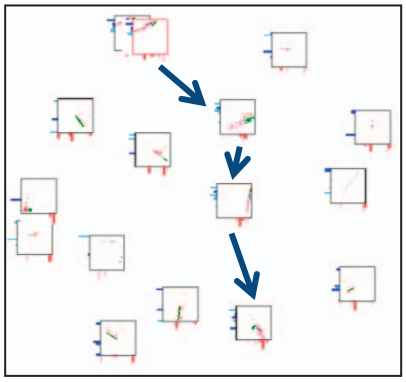


Fig. 10. Planning an exploration tour (indicated by arrows) in subspace S_5. Via the bar chart representation, glyphs in the same subspace neighborhood can be readily identified. The tour is conducted via the touchpad-based local sight explorer and the red-framed glyph contains the currently generated projection scatterplot there. The user is free to save this glyph as a snapshot into the map for a later revisit.

one variable at a time, with another one fixed. Our touchpad provides a more flexible, holistic interface—it is essentially an N-D slider. Any subset of dimensions can simultaneously influence the scatterplot projection, and the interactive touchpad makes this interaction very direct. It unifies all dimension selectors into one interface and allows a direct balance and tradeoff of data factors in the visualization. An important and unique visual aid in this undertaking is our motion trails that freeze the movements in time such that their magnitude and extent can be easily appreciated (see Fig. 8d). Fig. 9, on the other end, provides a multifactor (here 7-10) visualization with a single scatterplot, allowing users to aggregate factors along two or more orientations to express (or ignore) the effects of tradeoffs. This is also a unique feature of our system.

6.2 High-D Subspace Cluster Sculpting

The second usage scenario operates within a subspace-clustering scenario, using the image segmentation data set from the UCI Machine Learning Repository [32]. We took 1,200 instances composed of 300 random instances each from four classes (Brickface, Cement, Foliage, and Grass). Each instance corresponds to a 3×3 image region with a feature vector of 19 attributes (dimensions). These attributes are statistical measures of the images, such as region-centroid, region pixel count, density, hue, and others; for a complete listing see [32]. The goal is to determine descriptive feature vector clusters and from it derive models that can classify new image pixels into these classes.

For our experiment, we did not retain the class information of the data set since we seek to demonstrate the interactive semi-supervised subspace clustering capabilities of our framework. In the following, we will use the tourism metaphor that is at the core of the $\mathrm{TripAdvisor}^{\mathrm{N-D}}$ framework to illustrate the five exploration tasks. We first describe the implementation of the five tasks in detail and then present their use with the image segmentation data set.

Identify the sights (task 1). We first construct attractive initial destinations from which to start explorations. In this particular application, we use ENCLUS to find interesting subspaces that embed good clusters. In the following, we shall denote a dimension as X_i and a subspace as S_i. In this particular example, we identified five subspaces:

$$S_1 = (X_1, X_7, X_8, X_9, X_{18}), S_2 = (X_6, X_9, X_{18}, X_{19}),$$
$$S_3 = (X_7, X_9, X_{10}, X_{11}), S_4 = (X_8, X_{16}, X_{18}),$$
$$S_5 = (X_7, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}).$$

We then apply the projection pursuit algorithm (Section 4) to identify 15 views (the sights) from these subspaces and insert them into the sight map shown in Fig. 1.

Plan and go on the tour (tasks 2/3). The sight map now becomes the "tour map." Each view is represented as a sight glyph (Fig. 3). By clicking on one of the "destinations" in this map, its frame color changes from black to red (Figs. 10 and 11) and the corresponding projection view is shown as a scatterplot in the N-D sight explorer for closer exploration (see task 4: Hop off the bus, below). In this way, users may examine any sight in the tour map interactively using this interface, and in any order. But they may also use the distance and orientation information to connect the sites in some predefined order, allowing "tour designers" to plan an exploration tour either for themselves or for some "traveler" (as shown in Fig. 10). So, unlike a travel with the Grand Tour, analysts now have a map by which they can compare the N-D orientations of the projections and draw conclusions from their spatial associations. In practice, these steps are often revisited after gaining insights about certain landmark sights, whereby new snapshots of existing or new sights may be added along the way.

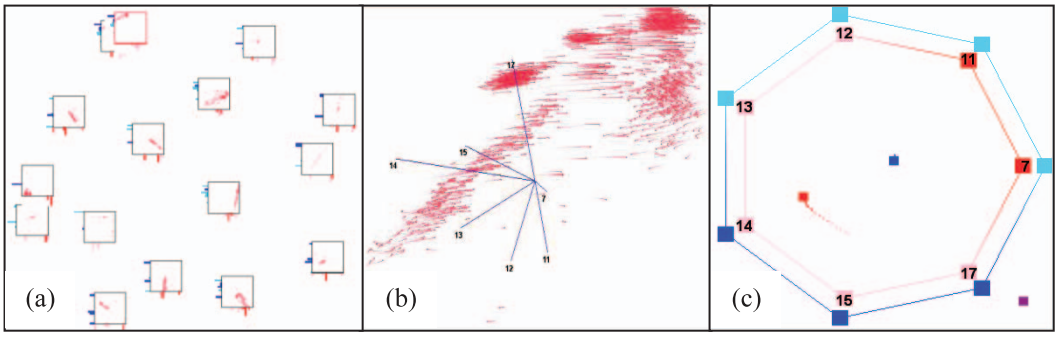


Fig. 11. Exploring a sight. (a) sight map—the location of the current view (as modified in the sight explorer) is shown red-framed as an orientation aid; (b) dynamic scatterplot with motion trails enabled, to provide an additional navigation hint and give a sense for cluster extent; (c) touchpad configuration for the view shown in (b).

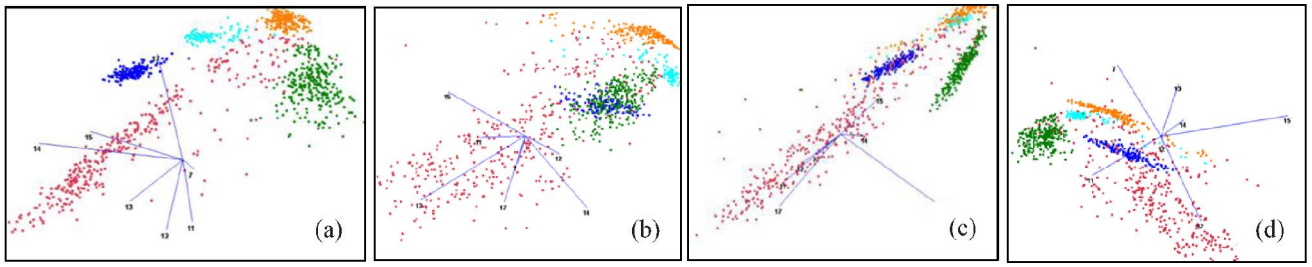


Fig. 12. (a) Brushing clusters discovered in the touchpad-based local exploration (see Fig. 11b). (b)-(c): Examining the brushed clusters in the different S_5 landmark views along the tour specified in Fig. 10. The clusters are separated quite well in all views. No new clusters are found.

Hop off the bus and explore the locale (task 4). Having found a potentially interesting destination (landmark or snapshot), the traveler may "hop off the bus" and look around the local neighborhood using the N-D touchpad-based explorer. Since each destination is a projection in a subspace, the touchpad interface only has the dimensions of that subspace, but dimensions can be added or removed at any time. We store the *PPA* vectors along with the destination projection view as well as the touchpad's vertex ordering. With this setup, the user can quickly start the tour exactly from the viewpoint of the selected destination.

Gain orientation (task 5). To enable global localization when controlling the travel with the touchpad, we show the current location of the view in the tour map as a red-framed sight icon of the view (see Fig. 10). This map location is computed by running MDS on the layout with the new view included. We have found that since the number of sights is typically fairly small, the time delay is negligible and the remaining icons reposition very little, if at all.

Usage scenario. In this example we follow Pam, a computer vision researcher who routinely uses visual cluster analysis to 1) derive models for recognizing features in images from local descriptors, and 2) use these models to categorize (label) images for content-based image retrieval (CBIR). Starting from the initial map in Fig. 1 found via projection pursuit Pam begins her exploration by picking the 7D subspace S_5 for closer inspection. She specifies a tour to examine the different adjacent views (see Fig. 10). Sight glyphs derived from the same subspace can be quickly identified by the similar distribution of dimension magnitude bars at the icon frame. Note that dimensions that are not in the subspace are represented as white colors in the icon. The scatterplot thumbnail already provides some insight into the type and structure of the cluster. Pam begins the tour with the sight icon shown on the top-left in the map. The touchpad interface will then be configured into an equilateral 7-gon. Fig. 11 shows a configuration from this interactive touchpad-based exploration. The relative location of the currently observed scatterplot is shown as a red-framed sight glyph in the map as an orientation hint. In the exploration Pam moves the PPA pointers across the interior areas of the touchpad polygon. This exposes multivariate relationships among the feature components and so reveals possibly interesting interactions among them. From the sequence of scatterplots so generated, Pam observes that there are three to five subclusters which are coherently moving together. These subclusters are likely due to nuanced feature populations, and including descriptors for these into the classification model can potentially lead to more accurate CBIR down the road.

To investigate her discovery further, she takes the best view (Fig. 11b) and brushes the data points in each of the separate clusters with different colors (Fig. 12a). She then continues on the tour using the touchpad (Figs. 12b and 12c) and finds that the brushed subclusters remain well separated and no new subclusters are found. She ends this exploration by acquiring a snapshot of the best view on these subclusters (Fig. 12a) and inserting it as a glyph into the tour map. This triggers the MDS algorithm to layout the space with the new view in the set. Note that our layout mechanism seeks to preserve the present layout as much as possible and typically the layout changes very little.

Discussion. This usage scenario started out with the results of a well-established subspace clustering algorithm (ENCLUS) which our fictitious user then refined via our framework. On the other hand, our local sight explorer started out with the results of a fairly sophisticated projection optimizer (projection pursuit), which our user subsequently refined by ways of our local sight explorer. It is clear that without these automatically acquired initial configurations obtaining these results would have consumed a vast amount of time. Therefore, the proposed symbiosis of pairing automated algorithms with a targeted visual refinement interface seems to be a winning strategy. Both components of our framework proved important in this effort: 1) the global sight map enabled the user to keep track of the various subspace clusters and the set of good projections, and 2) our local sight explorer enabled the user to refine these computed views in a multivariate context to best expose and confirm the underlying cluster structures and the interactions among their attributes.

Finally, we also compared our user-in-the-loop framework with the popular unsupervised k-means algorithm. Using the standard elbow strategy we found $k = 5$ to be a good choice. Fig. 13 shows the clustering result obtained with our framework and that with k-means, again using the image segmentation data set. We observe that while the Brickface data points (0∼300) are classified into two big clusters (rows 1 and 2) using our interactive framework, the automated k-means algorithm classified them into three clusters (rows 1, 2, and 5). For all other classes, except Grass (this class is well classified into a single cluster in both methods), TripAdvisor^{N-D} gives a clearer and better classification result. This is confirmed by a precision/recall analysis. Here, $\text{precision} = \text{TP}/(\text{TP} + \text{FP})$ while $\text{recall} = \text{TP}/(\text{TP} + \text{FN})$, where TP, FP, and FN are the numbers of true positives, false positives, and false

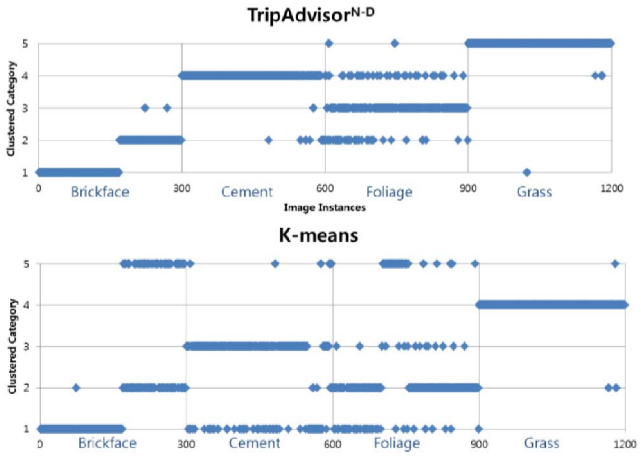


Fig. 13. Analysis of the clustering results obtained from the image segmentation data set. We observe that by putting the user into the loop (with our visual $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ framework) better clustering results than with automatic k-means can be obtained (we used $k = 5$ which gave the best results). $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ enabled clearer and more defined classification results for all classes but Grass, which was well defined for k-means as well.

negatives in that order. Table 3 shows that $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ enabled near-perfect precision (0.98) for 3/4 categories, while overall it enabled substantially better results ($>10\%$) than k-means in 4/8 instances and did similar in the remaining four instances.

7 Conclusions

We believe that our framework fills an important void in high-D data exploration as it provides both overview and detail within one synergistic and interactive interface. Further, by adopting a familiar paradigm—sightseeing as a real-life activity—for the data and space navigation a better understanding of high-D relationships can also be achieved. The two application examples we used—the multivariate preference-guided selection of optimal data configurations taking into account tradeoffs and the user-assisted space partitioning and cluster analysis—readily demonstrate how intuitive these popular and important tasks can be made by providing an interactive interface with direct controls and illustrative graphics. Via our system, analysts can consider more than three criteria/dimensions simultaneously allowing complex tradeoffs to be readily recognized, and they can sculpt odd-shaped clusters that do not fit the assumptions of automated clustering algorithms. Finally, we believe that by giving the user the ability to interactively "chase" preferred data configurations—using motion trails to visualize the trajectories—the popular biplots can also become much more intuitive, justifiable, and trustable.

We have demonstrated our system on data sets with relatively low dimensionality (less than 20D), but these types of data sets occur quite frequently in practice [2]. Both the sight glyph charts and especially the touchpad explorer, but not so much the sight map, are prone to scale issues. For the touchpad, a high dimensionality will require a vast number of vertex permutations to access all subspaces of the data. However, subspace analysis can often isolate local clusters with far fewer dimensions than the overall data space. We are also currently linking $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ with the dimension-space routing interface of [30] (see Section 5.1.3). It has a multiresolution dimension "zooming" capability which will enable an intuitive dimension reduction.

TABLE 3
Comparing Precision and Recall of $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ with that of k-Means (the Latter Is Shown in Parentheses)

Dataset	Precision	Recall
Brickface	0.99 (0.59)	0.56 (0.56)
Foliage	0.98 (0.70)	0.71 (0.62)
Cement	0.86 (0.92)	0.95 (0.72)
Grass	0.98 (0.99)	0.98 (0.98)

Our user base will likely not be casual users as yet, although in a rapidly growing world of interactive apps the visualization literacy continues to improve. Since our framework employs rather simple representations for visualization—bar charts and scatterplots—the visual language is not too unfamiliar to users somewhat literate in graphical plots. The interaction paradigms and system responses also appeal to familiar concepts such as motion parallax, map navigation, and selection by proximity. Extensive user studies will tune the system for general consumption.

Finally, ongoing work integrates $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ into the daily workflow of a team of collaborating climate scientists, who have been routinely using our prior ClusterSculptor system [17] to identify clusters of aerosol species in 450D mass spectra of millions of particles [29]. Since automated clustering tools proofed ineffective for their complex and noisy high-D data, having our visual tools allowed them to readily inject their domain knowledge-informed intuition into the cluster analysis process and so derive superior data models. $\mathrm{TripAdvisor}^{\mathrm{N\text{-}D}}$ is a significant advance since it enables cluster sculpting directly in N-D and not by tedious tuning of individual dimension weights in the 1D spectrogram augmented by a 3D PCA display [17].

Acknowledgments

This work was funded partly by US National Science Foundation (NSF) grants 1050477 and 0959979. The authors are also much indebted to Kevin T. McDonnell for proofreading the manuscript.

References

[1] D. Asimov, "The Grand Tour: A Tool for Viewing Multidimensional Data," *SIAM J. Scientific and Statistical Computing*, vol. 6, no. 1, pp. 128-143, 1985.
[2] C. Chabot, "Practical Applications of Visual Analytics: On the Cusp of Widespread Adoption," *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, 2008.
[3] C. Cheng, A. Fu, and Y. Zhang, "Entropy-Based Subspace Clustering for Mining Numerical Data," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 84-93, 1999.
[4] D. Cook, A. Buja, J. Cabrera, and C. Hurley, "Grand Tour and Projection Pursuit," *J. Computational and Graphical Statistics* vol. 4, pp. 155-72, 1995.
[5] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the Dice: Multidimensional Visual Exploration Using Scatterplot Matrix Navigation," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1539-1148, Nov./Dec. 2008.

[6] J. Friedman and J. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Trans. Computers,* vol. C-23, no. 9, pp. 881-890, Sept. 1974.
[7] K. Gabriel, "The Biplot Graphic Display of Matrices with Application to Principal Component Analysis," *Biometrika,* vol. 58, no. 3, pp. 453-467, 1997.
[8] S. Garg, J. Nam, I.V. Ramakrishnan, and K. Mueller, "Model Driven Visual Analytics," *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST),* pp. 19-26, 2008.
[9] S. Garg, I.V. Ramakrishnan, and K. Mueller, "A Visual Analytics Approach to Model Learning," *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST),* pp. 67-74, 2010.
[10] G. Hatfield, "Perception as Unconscious Inference," *Perception and the Physical World: Psychological and Philosophical Issues in Perception,* D. Heyer and R. Mausfeld, eds., pp. 115-143, Wiley, 2002.
[11] A. Inselberg and B. Dimsdale, "Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry," *Proc. First IEEE Conf. Visualization,* pp. 361-378, 1990.
[12] D. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An Interactive System for PCA-Based Visual Analytics," *Computer Graphics Forum,* vol. 28, no. 3, pp. 767-774, 2009.
[13] E. Kandogan, "Star Coordinates: A Multi-Dimensional Visualization Technique with Uniform Treatment of Dimensions," *Proc. IEEE Information Visualization Symp. Late Breaking Topics,* pp. 9-12, 2000.
[14] J. Hartigan, "Printer Graphics for Clustering," *J. Statistical Computation and Simulation,* vol. 4, no. 3, pp. 187-213, 1975.
[15] J. Kruskal and M. Wish, *Multidimensional Scaling.* Sage Publications, 1977.
[16] M. Meyer, H. Lee, A. Barr, and M. Desbrun, "Generalized Barycentric Coordinates on Irregular Polygons," *Graphics Tools,* pp. 1086-7651, 2002.
[17] J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre, "ClusterSculptor: A Visual Analytics Tool for High-D Data," *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST),* pp. 75-82, 2007.
[18] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: A Review," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 90-105, 2004.
[19] J. Seo and B. Shneiderman, "A Rank-by-Feature Framework for Unsupervised Multidimensional Data Exploration using Low Dimensional Projections," *Proc. IEEE Symp. Information Visualization,* pp. 65-72, 2004.
[20] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," *Proc. IEEE Symp. Visual Languages,* pp. 336-343, 1996.
[21] M. Sips, B. Neubert, J. Lewis, and P. Hanrahan, "Selecting Good Views of High-Dimensional Data Using Class Consistency," *Computer Graphics Forum* vol. 28, no. 3, pp. 831-838, 2009.
[22] R. Smith, R. Pawlicki, I. Kókai, J. Finger, and T. Vetter, "Navigating in a Shape Space of Registered Models," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1552-1559, Nov./Dec. 2007.
[23] N. Snavely, S. Seitz, R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 835-846, 2006.
[24] D. Swayne, D. Lang, A. Buja, and D. Cook, "GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization," *Comp. Statistics & Data Analysis,* vol. 43, no. 4, pp. 423-444, 2003.
[25] J. Talbot, B. Lee, A. Kapoor, and D. Tan, "EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers," *Proc. ACM Conf. Human Factors in Computing Systems (CHI),* pp. 1283-1292, 2009.
[26] S. Teoh and K. Ma, "PaintingClass: Interactive Construction, Visualization and Exploration of Decision Trees," *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD),* pp. 667-672, 2003.
[27] C. Ware, *Information Visualization: Perception for Design,* second ed. Morgan Kaufmann, , 2004.
[28] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, and W. Ribarsky, "Value and Relation Display: Interactive Visual Exploration of Large Data Sets with Hundreds of Dimensions," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 3, pp. 494-507, May/June 2007.
[29] A. Zelenyuk and D. Imre, "Beyond Single Particle Mass Spectrometry: Multidimensional Characterization of Individual Aerosol Particles," *Int'l Rev. in Physical Chemistry,* vol. 28, pp. 309-358, 2009.
[30] Z. Zhang, K. McDonnell, and K. Mueller, "A Network-Based Interface for the Exploration of High Dimensional Data Spaces," to Appear, *Proc. IEEE Pacific Visualization Symp.,* Mar. 2012.
[31] http://www.youtube.com/watch?v=3_qRwOyTD_w, 2012.
[32] UCI Machine Learning Repository (accessed 8/09) http://archive.ics.uci.edu/ml/data sets/Image±Segmentation, 2012.
[33] MAWI Working Group Traffic Archive (Accessed 8/08), http://mawi.wide.ad.jp/mawi/, 2012.
[34] College Prowler (Accessed 9/09), http://collegeprowler.com, 2012.
[35] US News Best Colleges (Accessed 9/09), http://colleges.usnews.rankingsandreviews.com, 2012.

Julia EunJu Nam received the PhD degree in computer science from Stony Brook University. Her research interests include visual analytics, visual data mining, and information visualization. She is currently a member of the Office Visio development team at Microsoft Corporation.

Klaus Mueller received the PhD degree in computer science from The Ohio State University and is currently a professor of computer science at Stony Brook University. His current research interests include visualization, visual analytics, and medical imaging. He won the US National Science Foundation (NSF) CAREER award in 2001 and the SUNY Chancellor's Award in 2011. He has authored more than 140 peer-reviewed papers. He is a senior member of the IEEE and the IEEE Computer Society. For more information, see http://www.cs.sunysb.edu/~mueller.