Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation

Niklas Elmqvist, *Member, IEEE*, Pierre Dragicevic, and Jean-Daniel Fekete

Abstract—Scatterplots remain one of the most popular and widely-used visual representations for multidimensional data due to their simplicity, familiarity and visual clarity, even if they lack some of the flexibility and visual expressiveness of newer multidimensional visualization techniques. This paper presents new interactive methods to explore multidimensional data using scatterplots. This exploration is performed using a matrix of scatterplots that gives an overview of the possible configurations, thumbnails of the scatterplots, and support for interactive navigation in the multidimensional space. Transitions between scatterplots are performed as animated rotations in 3D space, somewhat akin to rolling dice. Users can iteratively build queries using bounding volumes in the dataset, sculpting the query from different viewpoints to become more and more refined. Furthermore, the dimensions in the navigation space can be reordered, manually or automatically, to highlight salient correlations and differences among them. An example scenario presents the interaction techniques supporting smooth and effortless visual exploration of multidimensional datasets.

Index Terms—Visual exploration, visual queries, visual analytics, navigation, multivariate data, interaction.

1 Introduction

Developing novel visualization techniques has long been one of the core activities in information visualization research, and each installment of the annual conferences in the field introduces new visual representations for new types of data, each one more complex than the next. In light of this increasing complexity, *scatterplots* [8, 38] remain one of the oldest and simplest yet most flexible and widely used visual representations of them all. Employed in standard statistical data graphics as well as in commercial visualization tools such as Spotfire and Tableau (formerly Polaris [30]), scatterplots form part of the basic vocabulary of data visualization.

Scatterplots visualize multidimensional datasets by assigning data dimensions to graphical axes and rendering data cases as points in the Cartesian space defined by the axes. However, even if we employ 3D graphics as well as point color, shape, and size as graphical properties, a standard scatterplot diagram can only visually represent a handful of data dimensions at a time. Most real-world datasets have many more dimensions. Beyond choosing a more complex visual representation, the standard solution to this problem is to only visualize a subset of the dataset dimensions in the scatterplot, giving control of the visual mapping to the user. Unfortunately, this approach provides no clear structure to the visual exploration process of the dataset and does not explicitly show the relations between different subsets of data dimensions. On the other hand, alternative visual representations for multidimensional data—such as parallel coordinates [16], dense pixel displays [18], or dimensional stacking [22]—trade away some of the simplicity of scatterplots for more flexibility and visual expressiveness.

In this work, we are interested in preserving the simplicity and familiarity of scatterplots while addressing their shortcomings. We present a method based on scatterplots for visual exploration of multidimensional datasets using structured navigation in data dimension space, trading visual complexity for interactivity to reach a wider audience.

Following a review of existing work, this paper describes our scatterplot matrix navigation method as well as the purpose and motivation for the navigation techniques. We then describe how we integrate query sculpting functionality with the general scatterplot matrix metaphor. We also discuss dimension reordering to optimize grand tours of a dataset. We present a scenario using the tool for a multidimensional dataset and give some implementation details. The paper concludes with a summary and our plans for future work.

2 Background

In the following subsections, we will give a basic background to the challenges of multidimensional visualization and how previous work has attempted to overcome the complexity of high-dimensional datasets. We then introduce the concept of interactive exploration of data as a possible way of managing this complexity, and discuss how animated transitions can help in this endeavor.

2.1 Multidimensional Visualization

Keim [17] presents a taxonomy of primarily multidimensional visualization techniques, listing categories such as standard 2D/3D displays, geometrically transformed displays [7], iconic displays [6], dense pixel displays [18], and stacked displays [22]. Scatterplots are part of the first category, and are basic building blocks in statistical graphics and data visualization [8, 38]. Here, data cases are drawn as points in the Cartesian space defined by two or three graphical axes, their positions determined by the data dimension assigned to each axis.

The case for employing scatterplots for multidimensional visualization lies in their relative simplicity in comparison to other multidimensional visualization techniques, familiarity among users, and their high visual clarity [33]. Multidimensional visualization tools that feature scatterplots, such as Spotfire, Tableau/Polaris [30], GGobi [31], and XmdvTool [40], typically allow mapping of data dimensions also to graphical properties such as point color, shape, and size. Still, the number of dimensions that a single scatterplot can reliably visualize is considerably less than many realistic datasets.

To remedy this, scatterplot visualizations often give control of mappings from data dimensions to graphical properties directly to the user, allowing the user to dynamically change the visualized dimensions. However, this approach yields very little structure to the visual exploration and provides no relations between the data dimensions. Instead, multiple plots can be arranged in a scatterplot matrix [8] with the data dimensions on the rows and columns and each cell representing an individual scatterplot (see Figure 2 for an example scatterplot matrix for a simple dataset). While this approach does lend overview and structure to the exploration, the individual scatterplots in the matrix can appear a little like a set of image thumbnails with no indication how data points distribute between scatterplots when compared pairwise.

Our approach in this paper is to build on this very simple visual representation by adding structure and defining a navigation space for

- *Niklas Elmqvist is with INRIA in Paris, France, E-mail: nickelm@acm.org.*
- *Pierre Dragicevic is with INRIA in Paris, France, E-mail: dragice@lri.fr.*
- *Jean-Daniel Fekete is with INRIA in Paris, France, E-mail: jean-daniel.fekete@inria.fr.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

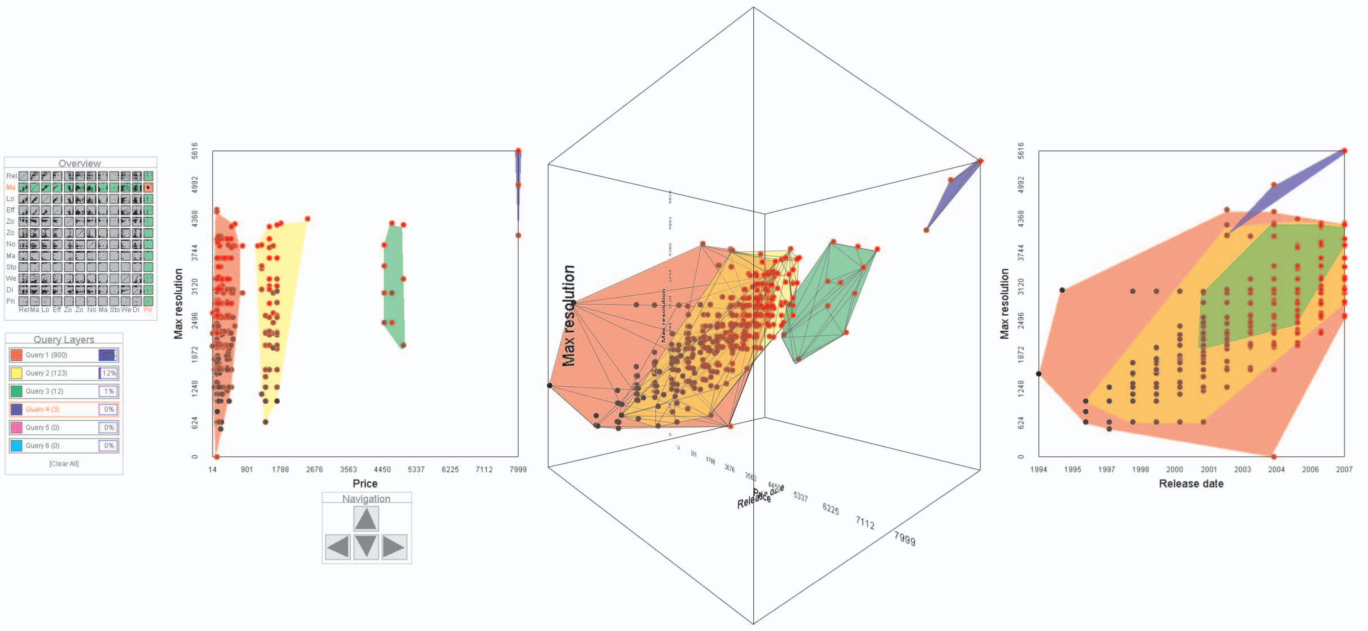


Fig. 1: Scatterplot matrix navigation for a digital camera dataset. The user is building queries for maximum camera resolution against price ranges and then studies them in relation to release year. The transition is performed using an animated 3D rotation.

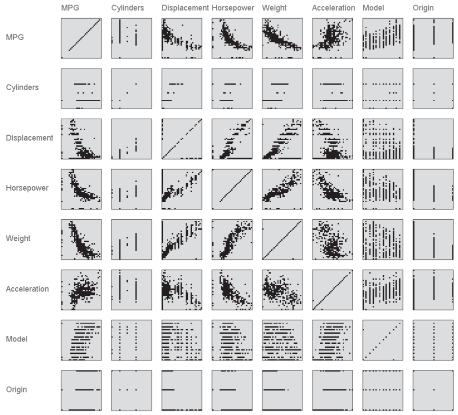


Fig. 2: Example of a scatterplot matrix for a 7-dimensional car dataset.

exploring the data. A similar approach could be employed for other simple visual representations, such as sets of one-dimensional visualization widgets.

2.2 Grand Tour

As we have seen, high-dimensional datasets pose a problem for 2D scatterplots. To visit such dataset using a scatterplot limited to showing two dimensions at a time, Asimov describes a method called the "grand tour" [4]. His idea is that a multidimensional dataset is fully visited when all the possible 2D projections have been seen, ignoring the symmetries and rotations. Therefore, a grand tour should visit a dense subset of all the possible 2D projections. This set is built according to criteria such as density, uniformity, continuity, linearity of the motion path, and some degree of user control.

Asimov describes several methods to compute approximations of a grand tour. However, Huber shows that it would require prohibitive time to extensively explore a high-dimensional dataset [15] so he advocates the projection pursuit method [21]. Projection pursuit has been designed to only show "interesting" aspects of high-dimensional spaces and has been adapted for visualization of such data [9].

2.3 Finding Interesting Planes

Several statistical methods have been designed to find the most interesting 2D planes and to create 2D views that best summarize a high-dimensional dataset; they are globally called *dimensional reduction* methods. Among these methods, Principal Component Analysis (PCA) is the simplest and most popular. The axis that maximizes the variance of all the projected points of the dataset is called the first component. The second component is orthogonal to the first and maximizes the variance of the projected dataset and so forth for all the other components up to n for an n-dimensional dataset. The plane defined by the first two components is frequently considered as the "most interesting" plane since it provides the best overview of the dataset in the sense of the largest variance. Furthermore, since PCA only considers linear projections, it fits well with the grand tour and projection pursuit ideas of important planes, whereas some other dimensional reduction methods find non-linear projections.

However, PCA suffers from several pitfalls, as described by Koren [20]: it is very sensitive to outliers and to artifacts in the data. Therefore, Koren proposes several variants of PCA that correct some of the issues but introduce parameters. Overall, all of these statistical methods find 2D planes. but have a very specific and restricted interpretation of the notion of "interest" and thus provide no guarantee that all the interesting planes will be found, nor that all planes found will actually be interesting.

Furthermore, and more importantly for our purpose, methods that compute 2D projections of high-dimensional spaces are beyond the understanding of casual information visualization users. As explained by Matthew Ericson during his keynote address at IEEE InfoVis 2007, scatterplots are considered too difficult to understand for readers of the New York Times, except when one of the axes is time. Understanding general scatterplots when the axes are meaningful is one thing, but understanding the meaning of axes that are linear combinations of attributes is beyond the skills of most people. Therefore, much research has focused on optimizing the navigation and selection of attributes.

2.4 Dimension Reordering

A simpler version of selecting "interesting" 2D planes is finding a rational order for exploring the dimensions of a dataset. Ankerst et al. [3] approach dimensional reordering from the viewpoint of placing similar dimensions close to each other, deriving a set of similarity metrics for general data and proposing an algorithm based on a translation of the problem to a Traveling Salesman Problem (TSP). Rosario et al. [26] take advantage of the special properties of nominal data to find an optimal dimensional mapping that minimizes visual clutter. Similarly, Peng et al. [24] generalize this to study visual clutter for a number of visualization techniques—scatterplots being one—and derive visualization-specific clutter measures. The rank-by-feature framework [28] presents a set of ranking criteria to the user and then makes the ranking of each projection of the dataset explicit, allowing the user to select among them.

2.5 Interactive Exploration

Interaction is a powerful tool for managing the complexity of high-dimensional datasets, particularly when the analyst has little or no existing information about the dataset, or when the exploration goals are unclear. More formally, visual exploration [17, 19] is the use of interactive visualization for exploratory data analysis [7, 8, 34]. Some existing work consider interactive data exploration in scatterplots. The original PRIM-9 system developed by Tukey et al. [35] allows for visualization of datasets of up to nine dimensions using a combination of projection and rotation followed by isolation and masking of the data into subsets.

High-dimensional filtering is a standard feature of information visualization systems through the use of dynamic queries [1, 41] that allow for direct manipulation of conjunctive filters; for instance, they form a cornerstone of the Spotfire application suite. However, standard dynamic query controls are not integrated into the visualization.

Some systems support the integration of query and filtering operations as well as their feedback directly into the visualization. High-dimensional brushes [23] support selection and filtering in data space using visual queries. Theron [32] integrates axis-filtering [27] in a a special-purpose parallel coordinate display. The DataMeadow [10] is a canvas for analysis of multidimensional data using a visual query language that provide filtering in high-dimensional space. The TimeSearcher [14] use box-queries in time-series visualizations where the queries are visualized as rectangles on top of the visualization.

2.6 Animated Transitions in Visualization

According to the principle of congruence [36], sudden changes in the layout of a visualization are disruptive since it prevents users from tracking changes over time. Animation can help solve that problem, although its utility for conveying complex information is uncertain.

Animation has long been an important part of data visualization; the rotation component of the PRIM-9 [35] system is central to understanding the structure of the 3D scatterplots visualized in the system. Other examples include the animated rotations in cone trees [25] that are used to show selection, the use of motion as a display dimension of its own in visualization [5], and the animated transitions and state changes in the Many Eyes [39] system.

Recently, Heer and Robertson [13] developed design guidelines for animated transitions between different data graphics as well as for value changes. While their transitions are basic interpolations of size, shape, and position, one key aspect of their approach is the use of stages in the animation to aid cognition. It seems that animated transitions are always better than instantaneous transitions when the motion of points is clearly understood. This is the case when changing one axis of a scatterplot: the points only move along one axis, so such animated transitions are easy to understand.

3 Scatterplot Matrix Navigation

The purpose of this work is to support structured visual exploration of multidimensional data using scatterplots. Instead of letting the user choose mappings for the axes of a single scatterplot, we create one scatterplot per every combination of dimensions and arrange them in a large *scatterplot matrix* [8, 35]. The whole matrix serves as an overview of the dataset and also defines a visual space for navigation, turning the visual exploration process into a navigation task. Transitions from one scatterplot to another is performed using a 3D rotation that is consistent with the overall navigation metaphor and that provides more natural cues than standard interpolated animation. Furthermore, we provide advanced visual queries using bounding volumes that allows the user to iteratively refine a query from different viewpoints using a set of query sculpting operations. Naturally, the order of columns and rows in the scatterplot matrix is significant, and can either be computed automatically as a function of the similarity between dimensions [3] or the degree of visual clutter [24], or manually by the user through drag-and-drop of rows and columns in an interactive mode [30, 40].

The primary tasks that we aim to support with our method is comparison and correlation of data dimensions in a dataset. These are compound tasks that consist of several smaller subtasks [2]. The animated transitions explicitly support these tasks by favoring object constancy and consistency between adjacent scatterplots, and the 3D nature of the transition carries more semantic meaning than naive interpolation. Furthermore, the highly interactive nature of the method is designed to promote direct and evocative control and overview of the data, supporting both reconfigure and connect interaction [42].

3.1 Visual Design

The two main visual components of our method consist of the current scatterplot and the scatterplot matrix. These components represent detail and overview of the dataset, respectively. The left side of Figure 1 shows this basic setup in our prototype implementation of the method.

The scatterplot component shows the currently viewed cell in the scatterplot matrix, together with the names and labels of the two displayed axes. The axes can have linear or logarithmic scales as controlled by the users. Data cases are drawn as points or boxes using a configurable colorscale that visualizes a user-selected dimension in the data.

The scatterplot matrix component (Figure 3) serves both as an overview and a tool for navigation. Miniature versions of the individual scatterplots are shown in the cells of the matrix, and the current position and potential navigation targets are highlighted using colors. The overview can also be used to show the extents of the visual queries (Section 4), a history of previous exploration, as well as suggestions for future exploration (see Section 5 for more details on how to use dimension reordering algorithms for this purpose).

3.2 Animated Transitions

As stated above, navigation in the scatterplot matrix is restricted to orthogonal movement along the same row or axis in the matrix. In other words, one dimension in the focused scatterplot is preserved while the other changes. The change is visualized using an animated transition. Instead of simply interpolating the position of each point for the transition, we perform the transition as a 3D rotation. This gives some semantic meaning to the movement of the points, allowing the human mind to interpret the motion as shape [37]. It also reinforces the metaphor of navigating in the space defined by the scatterplot matrix.

It is important to note that our technique is not based on 3D scatterplots. We use two dimensions for the visualization, and take advantage of the fact that the third dimension (depth) is hidden in the projection onto the 2D screen only for the actual transition to a new dimension

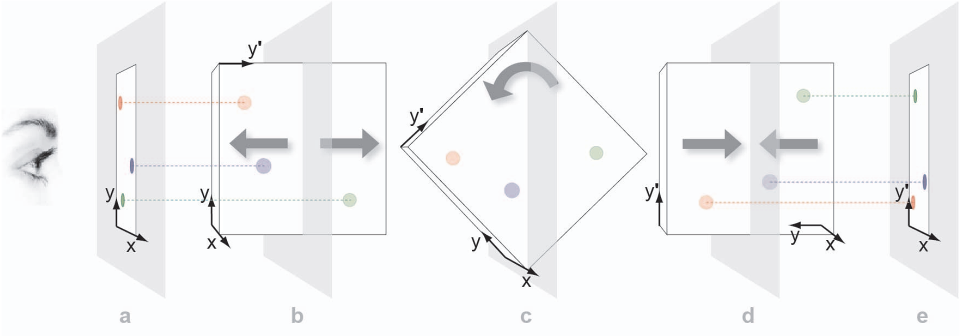


Fig. 4: Stage-by-stage overview of the scatterplot animated transition.

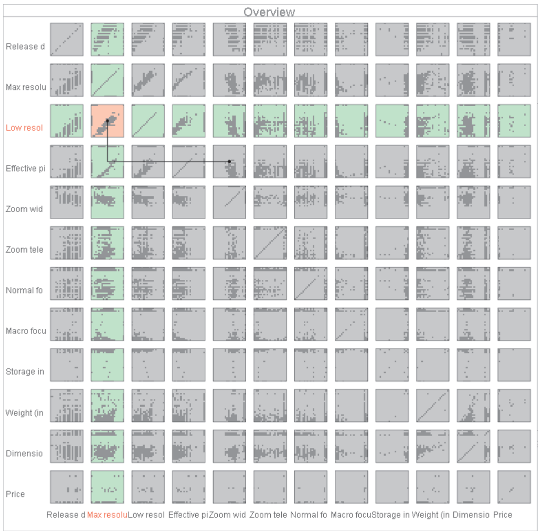


Fig. 3: Scatterplot matrix component used for overview and interaction in our prototype implementation.

assignment. In a way, this can be seen as a lazy or just-in-time allocation of graphical axes to data dimensions. Furthermore, the direction of the rotation—left or right for changing the horizontal axis versus up or down for the vertical axis–aids in the user's perception of their position and movement in the scatterplot matrix.

In accordance with the findings presented by Heer and Robertson [13], the actual transition is performed as a three-stage animation: extrusion into 3D, rotation, and projection into 2D. This is necessary for perspective 3D projection of the scatterplot due to perspective foreshortening.[1] More specifically, given two currently visualized dimensions x and y and a vertical transition to a new dimension y', these are the individual steps (see Figure 4 for a step-by-step visual sequence):

- **Extrusion:** The 2D scatterplot visualizing x and y is smoothly extruded to 3D where y' becomes the new depth coordinate for each data point. At the end of this stage, the 2D scatterplot has become 3D (Figure 4(a) and (b)).
- **Rotation:** The scatterplot is rotated 90 degrees up or down (depending on the relative position of y and y' in the scatterplot matrix), causing the axis previously pointing along the depth dimension to become the new vertical axis (Figure 4(c)).
- **Projection:** The 3D plot is projected back into 2D with x and y' as the new horizontal and vertical axes (Figure 4(d) and (e)).

3.3 Navigation

All navigation operations are restricted to orthogonal movement and are atomic transactions in that they never leave the main scatterplot visualization in an intermediate stage—when the user stops interacting, the visualization settles into one of the 2D scatterplots in the scatterplot matrix. This ensures that the navigation metaphor of navigating between 2D projections of the dataset is consistent. The user may still directly control the transition (using the "scratching" operation) in order to see details in the comparison and correlation of data points.

The interaction techniques supported in the framework have been designed to support all levels of the visual exploration task. Here follows a list of them (refer to Figure 5 for a graphical overview):

- **Stepping:** The user may utilize the arrow keys on the keyboard to step through the adjacent cells in the scatterplot matrix, giving the user direct and tangible control of the navigation in dimension space. The time the keyboard button is held down will govern the speed of the animated transition; a quick tap will result in a fast transition, a longer press will cause a slower one. The on-screen navigation bar (Figure 7(a)) also supports stepping in the same way, but is primarily intended for touch tablet interaction (see Section 7 for more information on this).
- **Scratching:** A little like crossfading between two records for hiphop music (also known as "scratching"), this interaction technique gives the user direct control of the transition by clicking the current matrix cell and moving the mouse towards the intended direction. The user can directly manipulate the animation (including pausing it and moving it back and forth) by dragging across the surface of the matrix overview. Releasing the mouse button will return the visualization to the nearest scatterplot.
- **Path planning:** Hovering with the mouse over the scatterplot matrix overview will show the shortest path from the current position to the indicated position in the matrix. Clicking will cause a sequence of animated transitions to be performed as the scatterplot visualization follows the planned path to the destination.
- **Path drawing:** Dragging the mouse on the matrix overview will draw a path on the surface of the scatterplot matrix. Once the button is released, the animation is performed in sequence.
- **Hyperjump:** Sometimes the user wants to jump to a different part of the scatterplot matrix without passing through the intermediate cells. Hyperjumping allows for a right-click on the overview matrix to perform this operation. Changing both dimensions at the same time will require two transitions, however; one horizontal and one vertical. It is important to note that, unlike simply changing the mapping of one or several axes for a standard scatterplot visualization, hyperjumping preserves the user's spatial cognition by making use of the scatterplot matrix.

[1] For orthographic (parallel) projection, the animation can be directly performed as a rotation with no need for extrusion or projection.

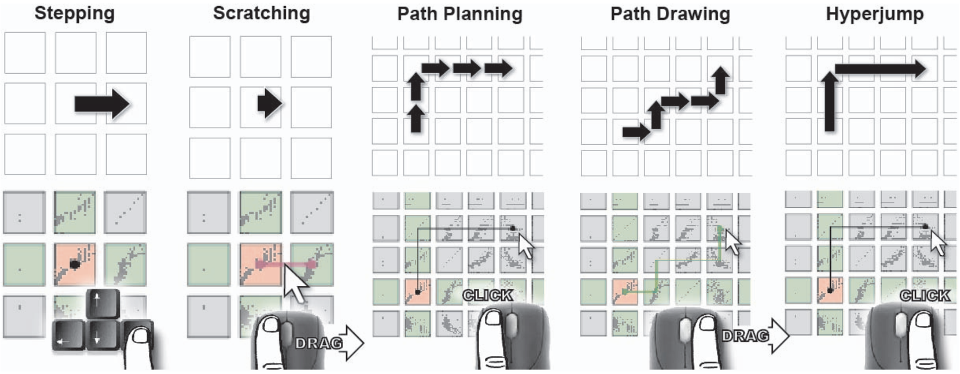


Fig. 5: Overview of the navigation operations supported by the scatterplot matrix navigation method.

3.4 Drill-Down, Reordering, and Overview Interaction

Beyond the navigation operations, we define several other interaction techniques. Drill-down to see details is central to visual exploration [29], and we support this by an implementation of the excentric labeling [12] technique as a resizable lens that can be moved on the surface of the main scatterplot (Figure 6).

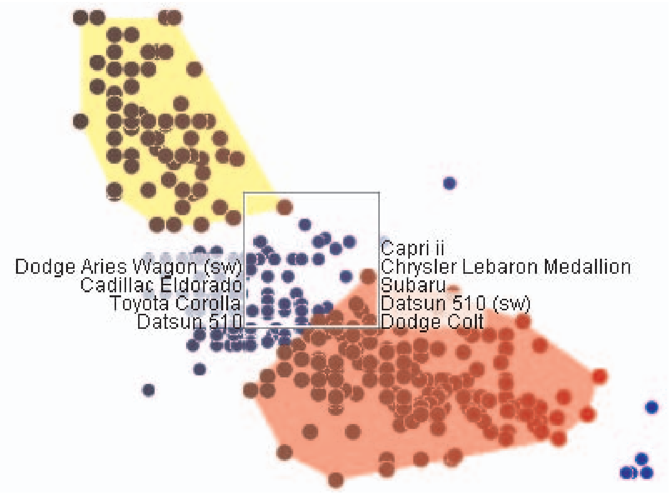


Fig. 6: Excentric labeling lens showing labels for selected points (indicated by hulls and a red colorscale; unselected points are blue).

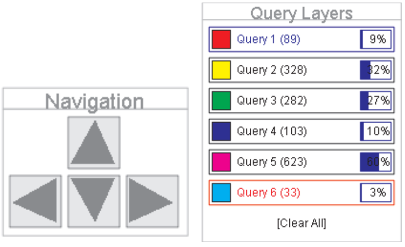
Another important feature is support for manual reordering of dimensions, akin to systems like Polaris [30] and XmdvTool [40]. Our prototype implements this through simple drag-and-drop operations where the user can grab a row or a column in the overview matrix and place it at the desired position.

Finally, holding down a key (currently the Control key in our prototype) animates the overview scatterplot matrix to fill the screen; releasing the key scales the matrix back down to its original size. This large version allows for better viewing of details in the scatterplot miniatures as well as for high-precision navigation (such as when scratching). The matrix is rendered using semi-translucency so that it still shows the scatterplot visualization in the background.

4 Query Sculpting

To facilitate visual exploration using the scatterplot matrix navigation method described above, we define an iterative filtering mechanism that we call *query sculpting*. Essentially, query sculpting allows for selecting data items in the main scatterplot visualization using 2D bounding volumes (boxes or convex hulls) and then iteratively refining the selection from other viewpoints while navigating the scatterplot matrix. The metaphor is a sculptor repeatedly chiseling away at the outline of a block of stone and rotating it until the sculpture is complete. Points can either be removed or grayed out. The technique is similar to high-dimensional brushing [23], but the iterative sculpting aspect is an integral part of our scatterplot matrix navigation method.

Our prototype implementation provides a number of color-coded queries that can be used during the visual exploration. A special query window (Figure 7(b))—modeled on the Photoshop image layers window—is used for selecting, naming, and clearing the queries. All query operations operate on the currently selected query, and clicking and dragging one query onto another will perform a *union* or *intersection* operation (by dragging using the left or right mouse button, respectively), similar to how layers are merged in Photoshop. Each query layer also gives a visual indication of the number and percentage of items currently selected by it.



Query Layers

🟥	Query 1 (89)	9%
🟨	Query 2 (328)	32%
🟩	Query 3 (282)	27%
🟦	Query 4 (103)	10%
🟪	Query 5 (623)	60%
🟦	Query 6 (33)	3%

[Clear All]

(a) (b)

Fig. 7: User interface components: (a) Navigation bar. Provides the same navigation functionality as the keyboard arrow keys and is especially suitable for interaction when using a touch tablet. (b) Query layer control box. Each line shows the query name and color as well the number and percentage of items selected by it.

Queries are sculpted in the data space of the current dataset, two dimensions at a time (the dimensions currently viewed in the scatterplot visualization) using *lasso* or *rubberband box selection* on the 2D surface of the visualization. The selection may either be a union or an intersection operation (again by using the left or right mouse button) with the existing items in the current query.

The visual representation of queries conforms to the overall navigation metaphor and is extruded, rotated, and projected in 3D as the user navigates the scatterplot matrix. This helps the user by creating a smooth transition between the extents of each query in two adjacent scatterplots. Furthermore, to support a tighter definition of the selection areas, we support both rectangular boxes as well as convex hulls to bound the selected items for each query layer.

5 Dimension Reordering

The concept of a *grand tour* [4] automatically portraying a high-dimensional dataset as a sequence of projections onto lower-dimensional subspaces is a potentially useful method for helping a user get an overview of a new dataset and see some of its salient features. In the context of our method, we could define the grand tour as an animated sequence of 2D scatterplot diagrams. However, in the interest of maintaining the interactive aspect of the visual exploration paradigm [17], we opt to instead reorder navigation space—i.e. the data dimensions in the scatterplot matrix—in such a way that salient features are highlighted, and continue to let users navigate themselves.

We adopt the systematic dimension reordering approach of Ankerst et al. [3], where similar dimensions in a multidimensional dataset are placed next to each other. Our implementation uses an external TSP implementation[2] to compute an optimal ordering. Such implementations are typically able to find an optimal ordering for datasets of less than 100 dimensions in a reasonable time (i.e. a few seconds at most).

The use of a scatterplot matrix allows us to have independent row and column orderings. In our method, it makes sense to utilize this fact and display *similarity* on the column and *dissimilarity* on the row order.

There are a number of suitable similarity measures for our problem depending on what similarity features to highlight; we use the absolute correlation of pairwise dimensions and its inverse for the dissimilarity metric D and the similarity metric S, respectively:

$$D_{X,Y} = |\rho_{X,Y}| \quad (1)$$

$$S_{X,Y} = 1 - |\rho_{X,Y}| \quad (2)$$

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (3)$$

In our prototype implementation, users can choose to reorder the dimensions of a dataset after loading it into the application. This will cause the navigation space to be arranged for structured visual exploration: horizontal movement explores scatterplot sequences of similar dimensions, whereas vertical movement shows dissimilar dimensions.

6 Implementation Notes

Our prototype implementation of scatterplot matrix navigation (ScatterDice) is built in Java and uses the InfoVis Toolkit [11] for all dataset management, loading, and reordering functionality. Rendering is performed using OpenGL through JOGL. Performance is interactive even for very large datasets with framerates of over 100 FPS.

The animated transitions are implemented using a three-stage sequence, as outlined above. The extrusion and projection steps simply assign data values from the new dimension to the depth (z) coordinate using an animated parameter as a weight, and the rotation is performed by rotating the 3D camera. Instead of keeping track of the camera orientation, moving to a new cell in the scatterplot matrix simply resets the camera to its default and swaps the axis mappings as needed.

Many implementation details that are not critical to the general idea of the framework remain open, including the choice of graphical marks (2D points, 3D boxes or 3D spheres), the use of perspective or parallel projection, and whether or not to start stepping animations immediately when the user presses a key or to wait until the keypress has finished. In our prototype, we have turned as many of these choices as possible into options configurable by the user.

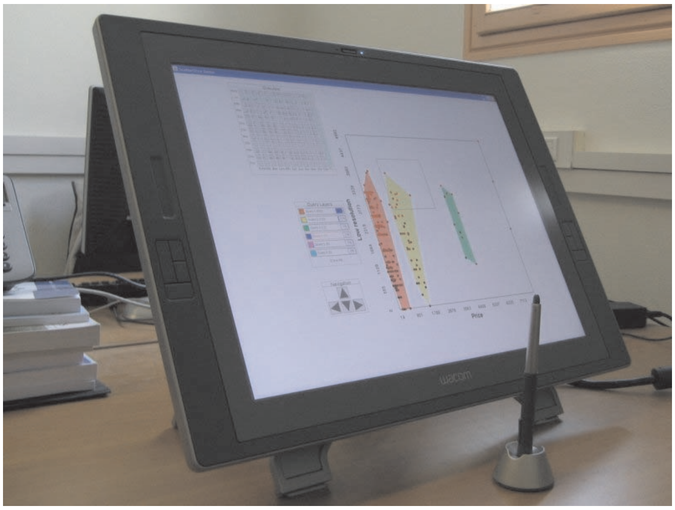


Fig. 8: Wacom CintiQ 21UX graphical tablet used for the visual exploration scenario, here shown running our prototype implementation.

7 Usage Scenario

We now demonstrate how our method could be used for visual exploration in scatterplot matrices. We will use a very basic scenario based on exploring a digital camera dataset in order to find a suitable camera given an initially loose set of constraints. The operations involved in this simple task are common to general multidimensional visual exploration for both expert and novice users alike.

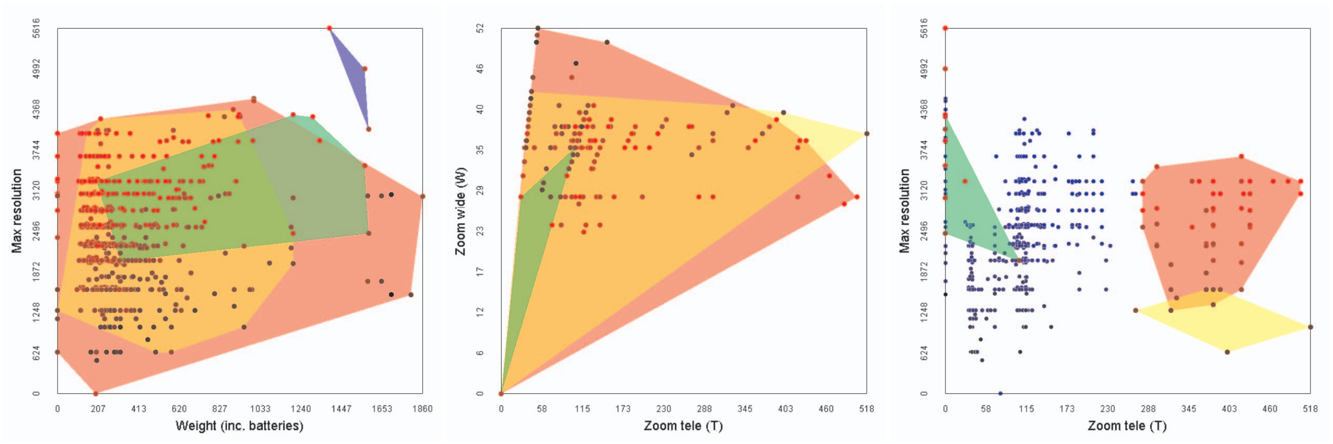
We will follow Liza, a visualization analyst who is trying to make an informed decision on buying a digital camera using our prototype implementation for multidimensional visual exploration. Liza uses a graphical tablet with our tool to make the interaction smooth and effortless (Figure 8). She uses a camera dataset built from online camera shopping and review sites.

Liza has no previous knowledge of digital cameras, and so her initial exploration goals are very fuzzy and she would like to let the visual exploration process itself guide and inform her selection. All she knows is her approximate budget—at most around $1000—and that she would like an all-round camera with good image resolution that she can use to take both nature shots as well as portraits.

Loading up a digital camera dataset containing 1039 cameras categorized in 13 dimensions (one being the name, the rest categorical or numerical) in the scatterplot navigation application, Liza is confronted with the main interface of the application. Since price and resolution is one of her most important constraints, she *hyperjumps* to the scatterplot showing maximum resolution on the vertical axis and price on the horizontal axis. She immediately notices that digital cameras seem to cluster into four different price ranges—between 0 and $1000, $1000–$3000, $3000–$6000, and $6000 and above—and she uses the *lasso tool* to create four different queries, one for each of the four clusters (see the leftmost plot of Figure 1). She notices that price does not appear to have much correlation with maximum resolution, since even the cheapest cameras span the whole range of resolutions—except for the highest price range query (the blue one, containing 3 of the total 1039 cameras in the dataset), that make up the high resolution range.

Trying to get a feel for the dataset, Liza *steps* to the right in the scatterplot matrix, giving rise to a smooth horizontal rotation (the center scatterplot in Figure 1) that finishes with the release year of the cameras on the horizontal axes of the scatterplot (rightmost in Figure 1). Here she immediately sees that the cheapest category of cameras (the red one, containing 900 cameras or 87% of the dataset) is spread over the whole range of release years. However, it appears that as digital cameras have matured, the newer and more expensive categories

[2] Concorde, see *http://www.tsp.gatech.edu/concorde/*



(a)

(b)

(c)

Fig. 9: Digital camera dataset visual exploration session: (a) weight distribution; (b) wide and tele zooms; (c) maximum resolution.

of cameras have appeared, with the most expensive category only appearing in 2002.

Liza knows that weight is an important factor for her choice, so she *drags and drops* the weight column to the right of the current column in the scatterplot matrix. She then *steps* to the right, bringing weight onto the horizontal axis through another smooth horizontal rotation. Figure 9(a) shows this scatterplot. It appears that the red (low-budget) and green (upper-budget) queries are spread all over the weight range, whereas yellow (mid-budget) are compact cameras with low weight, and the blue (high-budget) query are relatively heavy SLR-type professional cameras.

Intrigued by the mid-level cameras designated by the yellow query, Liza decides to change her exploration goals somewhat and incorporate both the red and yellow queries in her search even though the high $3000 range of the yellow query set is beyond her budget. Since she wants to be able to take both close-up portrait and long-range nature shots, she next navigates to the scatterplot showing the tele and wide zoom measurements through *path-planning*. This gives her some continuity to see how the four queries distribute as the tool smoothly animates to the new position in the scatterplot matrix. Figure 9(b) shows the destination scatterplot. Interestingly enough, the two higher price ranges (green and blue) do not have high zoom capabilities. Liza *drills down* into the dataset using the excentric label lens and realizes that this is because the green and blue cameras are primarily system cameras that have interchangeable lenses and come equipped with a very basic allround lens (if they have a lens at all).

Moving on, Liza *sculpts* both of the red and yellow queries to include only cameras with both high tele and wide zoom. She then *drags* the maximum resolution row to the position above the current row and then *scratches* up to bring the new dimension into view. Figure 9(c) shows the new view in the application. Disappointingly enough, the yellow mid-range query that caught Liza's eye previously all have low maximum resolutions. However, Liza now feels that she has enough knowledge of the dataset to perform her search in earnest.

Clearing all queries, Liza *hyperjumps* back to the nearest price dimension and uses the *lasso tool* to build a new red query containing both the low and mid-level budget cameras (1023 cameras). She then proceeds to *sculpt* her query through a succession of scatterplots, selecting only cameras with more than 3 megapixel resolution (489 cameras), then cameras that weigh less than 1000 grams (477 cameras), and finally cameras with more than 400 mm focal length (15 cameras). Navigating back to the maximum resolution plot, she refines the query to select only the two cameras with the highest megapixel capability. Figure 10 shows her resulting scatterplot on the price and tele zoom dimensions.

Drilling down into the dataset using the excentric label lens, Liza finds out that these two cameras are the Leica V-LUX 1 ($529) and the Panasonic Lumix DMC-FZ50 ($149). After having read reviews for both of these, Liza settles for the Leica as her camera of choice.

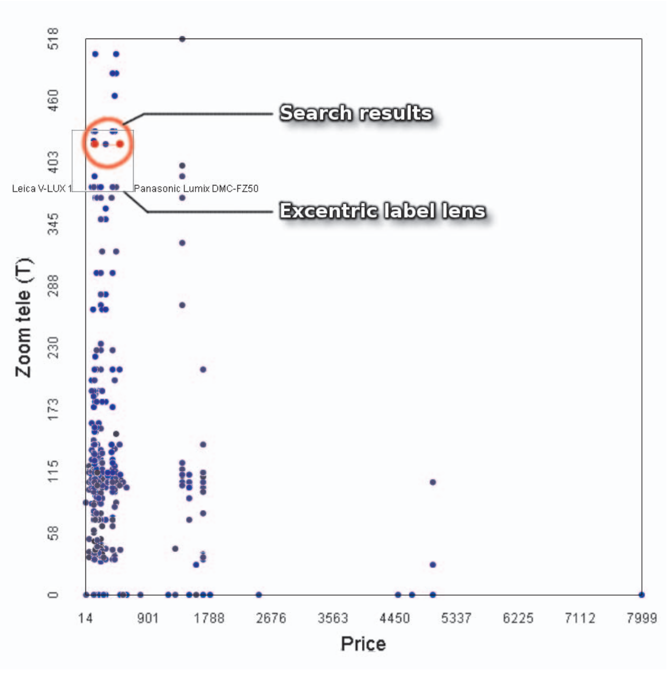


Fig. 10: Final visual exploration result for the digital camera dataset. The red circle denotes the two finalist cameras.

8 Conclusions and Future Work

In this paper, we have proposed a coherent design that uses scatterplots for exploring multidimensional datasets. Its benefit lies in the seamless integration of several components that all support the basic tasks of comparing and correlating dimensions in this kind of dataset. More specifically, the contributions of this paper are the following:

- A unified design for structured visual exploration of multidimensional data in scatterplot matrices using 3D animated transitions,
- A set of carefully designed interaction techniques for navigation in scatterplot matrix space,

- An approach for iterative refinement of queries by bounding volume sculpting that is integral to the scatterplot navigation metaphor, and
- A method for dimension reordering of scatterplot matrix rows and columns designed to show correlation and disparities, respectively, between individual dimensions in the dataset.

In the future, we envision validating this design through empirical evalution. User evaluation is difficult for such broad tasks as visual exploration, so we anticipate performing qualitative or longitudinal studies. We are also interested in investigating similar ways of turning a complex visualization into navigable sequences of simpler visualizations.

Acknowledgements

This work was supported in part by the joint Microsoft Research/INRIA ReActivity project. Thank you to Nathalie Henry for her comments on this manuscript.

References

[1] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the ACM CHI'92 Conference on Human Factors in Computing Systems*, pages 619–626, 1992.

[2] R. A. Amar, J. Eagan, and J. T. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 111–117, 2005.

[3] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 52–62, 1998.

[4] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, Jan. 1985.

[5] L. Bartram and C. Ware. Filtering and brushing with motion. *Information Visualization*, 1(1):66–79, 2002.

[6] H. Chernoff. Using faces to represent points in *k*–dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.

[7] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[8] W. S. Cleveland and M. E. McGill, editors. *Dynamic Graphics for Statistics*. Statistics/Probability Series. Wadsworth & Brooks/Cole, Pacific Grove, CA, USA, 1988.

[9] S. L. Crawford and T. C. Fall. Projection pursuit techniques for visualizing high-dimensional data sets. *Visualization in scientific computing*, pages 94–108, 1990.

[10] N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: a visual canvas for analysis of large-scale multivariate data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 187–194, 2007.

[11] J.-D. Fekete. The InfoVis Toolkit. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 167–174, 2004.

[12] J.-D. Fekete and C. Plaisant. Excentric labeling: dynamic neighborhood labeling for data visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 512–519, 1999.

[13] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visusalization and Computer Graphics*, 13(6):1240–1247, 2007.

[14] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[15] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, June 1985.

[16] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[17] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[18] D. A. Keim and H.-P. Kriegel. VisDB: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49, Sept. 1994.

[19] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proceedings of the Tenth International Conference on Information Visualization*, pages 9–16, 2006.

[20] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):459–470, 2004.

[21] J. B. Kruskal. Toward a practical method which helps uncover the structure of the set of multivariate observations by finding the linear transformation which optimizes a new 'index of condensation'. In R. C. Milton and J. A. Nelder, editors, *Statistical Computation*, pages 427–440. Academic Press, New York, 1969.

[22] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of the IEEE Conference on Visualization*, pages 230–237, 1990.

[23] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 271–278, 1995.

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

[25] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 189–194, 1991.

[26] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95, 2004.

[27] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35(7):80–86, 2002.

[28] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 65–72, 2004.

[29] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society Press, 1996.

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

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

[32] R. Theron. Visual analytics of paleoceanographic conditions. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, pages 19–26, 2006.

[33] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.

[34] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.

[35] J. W. Tukey, M. A. Fisherkeller, and J. H. Friedman. PRIM-9: An interactive multi-dimensional data display and analysis system. In W. S. Cleveland and M. E. McGill, editors, *Dynamic Graphics for Statistics*, pages 111–120. Wadsworth & Brooks/Cole, 1988.

[36] B. Tversky, J. B. Morrison, and M. Bétrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[37] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[38] J. M. Utts. *Seeing Through Statistics*. Brooks/Cole, 3rd edition, 2005.

[39] F. B. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. M. McKeon. Many Eyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[40] M. O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 326–333, 1994.

[41] C. Williamson and B. Shneiderman. The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 338–346, 1992.

[42] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions of Visualization and Computer Graphics*, 13(6):1224–1231, 2007.