DataMeadow: A Visual Canvas for Analysis of Large-Scale Multivariate Data

Niklas Elmqvist* INRIA/LRI, Univ. Paris-Sud
John Stasko† Georgia Institute of Technology
Philippas Tsigas‡ Chalmers University of Technology

Abstract

Supporting visual analytics of multiple large-scale multidimensional datasets requires a high degree of interactivity and user control beyond the conventional challenges of visualizing such datasets. We present the DataMeadow, a visual canvas providing rich interaction for constructing visual queries using graphical set representations called DataRoses. A DataRose is essentially a starplot of selected columns in a dataset displayed as multivariate visualizations with dynamic query sliders integrated into each axis. The purpose of the DataMeadow is to allow users to create advanced visual queries by iteratively selecting and filtering into the multidimensional data. Furthermore, the canvas provides a clear history of the analysis that can be annotated to facilitate dissemination of analytical results to outsiders. Towards this end, the DataMeadow has a direct manipulation interface for selection, filtering, and creation of sets, subsets, and data dependencies using both simple and complex mouse gestures. We have evaluated our system using a qualitative expert review involving two researchers working in the area. Results from this review are favorable for our new method.

Keywords: Multivariate data, visual analytics, parallel coordinates, dynamic queries, iterative analysis, starplot, small multiples.

1 Introduction

Managing and presenting large, high-dimensional datasets is one of the core problems in information visualization, and the vast number of different approaches to solving this problem attests to its difficulty [16]. However, to be able to support efficient visual analytics for such datasets we must also provide smooth and meaningful interaction techniques for selecting, filtering and combining the data. Furthermore, these techniques must be capable of operating on multiple large-scale datasets instead of just one, and must allow for communicating the results of the analysis to an outside audience at a later stage [30].

The method presented in this paper is called the DataMeadow (see Figure 1), and it provides users with a canvas for exploring multidimensional data sets using advanced visual queries. The data itself is represented by a DataRose, a color-coded, parallel coordinate starplot displaying selected variables of the set. Each displayed variable can be filtered using dynamic query bars [25, 34] present on each rose axis. Individual DataRoses are connected in a data flow fashion; these connections are illustrated by arrows exiting the center of one DataRose and entering the center of another, as illustrated in the figure. In this way, the user can progressively build more and more complex queries with varying subsets of the data being passed along.

*e-mail: elm@lri.fr
†e-mail: stasko@cc.gatech.edu
‡e-mail: tsigas@cs.chalmers.se

IEEE Symposium on Visual Analytics Science and Technology 2007
October 30 - November 1, Sacramento, CA, USA
978-1-4244-1659-2/07/$25.00 ©2007 IEEE

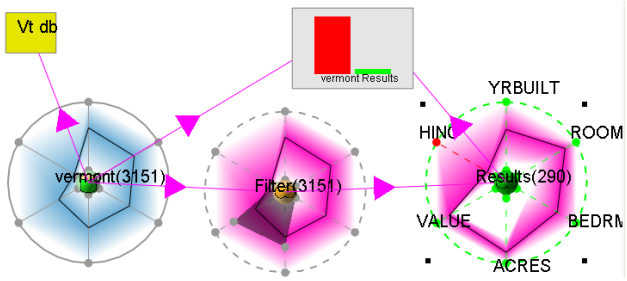

Figure 1: Sample house value and acreage versus number of rooms and owner income query in the DataMeadow.

Furthermore, the incrementally-refined queries can be annotated with various visual representations in order to communicate the results to stakeholders (i.e. communication-minded visualization [30]). For added flexibility, the roses can be freely moved around, resized, and manipulated on the meadow canvas to allow for easy comparison to other datasets. To provide for more complex comparisons, DataRoses come in different types, either representing a data source or a specific set operation such as union, intersection, or uniqueness. This allows roses to be connected to other roses using dependencies, forming visual query chains. In essence, the DataMeadow provides a form of "visual pivot table", allowing the user to refine and examine selected portions of a large multivariate data set in parallel.

In order to assess the utility and interaction efficiency of the method, we performed an expert review using a think-aloud protocol involving two visualization researchers. Our observations from this study indicate that the DataMeadow is a useful way of thinking and interacting with multivariate data. The participants both remarked on the ease of creating queries and the power of being able to "play" with the data and getting immediate feedback.

The rest of this paper is organized as follows: We begin with a tour of the existing work on visualization and visual analytics of multivariate data. We then formulate the requirements for an analysis tool intended for such data, including identifying the user group and the main user tasks. We describe the DataMeadow visual canvas in detail and describe a typical scenario using the tool. This is followed by our user evaluation and the results we gained from it. We finish the paper with a discussion and our conclusions.

2 Related Work

The work presented in this paper builds on ideas and inspiration both from techniques for visualizing multivariate data, as well as the application of these techniques to highly interactive interfaces for visual analytics. We describe both of these areas in turn in the following sections.

2.1 Multivariate Visualization

Much work has been conducted on visually presenting hypervariate data in a form suitable for understanding; Keim [16] presents

an overview and taxonomy of such techniques. For large sets of multidimensional data, standard 2D or 3D symbolic displays such as plots, diagrams, and charts are generally insufficient due to scalability reasons, and more advanced methods are needed. Examples of such methods include geometrically-transformed displays [8], iconic displays [7], dense pixel displays [15, 17, 18], and stacked displays [14, 20, 24].

One prolific geometrically-transformed display technique is parallel coordinates [12, 13], which abandons the standard practice of orthogonal dimension axes, and instead stacks up the axes in parallel, tracing a line instead of a point through the axes for each data case. The diagram is then easily extended with just another parallel axis for each new dimension that is to be visualized. To avoid a linear extension of the diagram *ad infinitum*, a so-called *starplot* is constructed where the diagram is folded into polar space, mapping each axis on the radius of a circle. The DataRose presented in this work is a direct descendant of the starplot and has indeed a parallel coordinate mode, but also other visual representations showing data distribution.

Fua et al. [10] introduce hierarchical parallel coordinates that are rendered in clusters using opacity bands instead of drawing each individual data point, just like in the DataRose. The approach was later extended to starplot displays. However, DataRoses are manually clustered by the analyst and also allow the use of histogram bands, thus providing a more faithful rendition of the underlying data than the mean and extreme values shown by the opacity bands.

The parallel sets technique [4] is another approach to representing distribution for categorical data in a parallel coordinate diagram. It uses proportional scales and color paths to show how different categories divide among adjacent dimensions. Sifer [27] extends the idea by removing the color paths and instead relies on implict color coding. The DataRose also makes data distribution in the parallel coordinate display explicit, but our approach does not require categorical or hierarchical data.

The DataMeadow presented here can support a very large number of data cases, but if the number of variables to visualize grows too large, the scalability of the technique is affected. In such cases, we must employ techniques for very high dimensional representation, such as the dense pixel displays and stacked displays mentioned earlier. Our datasets are not of this magnitude, but we can easily foresee integrating visual elements based on these visualizations onto our canvas as well.

2.2 Multivariate Visual Exploration

Interaction is a powerful means for multivariate data exploration. The Dust & Magnets [37] technique is an example of this, and shows how a simple interaction can provide important insights into a complex dataset through animation. Another example is the parallel coordinate tree [6] introduced by Brodbeck and Girardin for presenting hierarchical and multidimensional data using a tree representation. Their use of focus+context distortion for interacting with the visualization fulfills an integral role in the exploration of the data.

The Sandbox [35] system is a platform for visual analysis of integrated information in a semi-structured fashion. The tool emphasizes fluid interaction on a 2D canvas using direct manipulation in order to promote visual thinking, much like the DataMeadow canvas presented in this work. Towards this end, the Sandbox even supports a gesture detection component, just like our method. However, where the Sandbox uses unstructured or semi-structured visual elements, we impose a multivariate data model on our visual elements in order to allow for faithful visualization of the data.

Theron presents the concept of interactive parallel coordinate plots (IPCPs) [29] as an interactive tool for analysis, providing interaction techniques such as brushing [2] and axis filtering [23] similar to the DataRose approach in this paper. However, the DataMeadow allows for linking several DataRoses together to construct composite queries that are dynamically updated as the analyst interacts with the visual elements.

Finally, other work on visual analytics has tackled the problem of multivariate data: Brennan et al. [5] present a framework for exploration of multidimensional data and employ a visual canvas, but they focus on collaborative aspects of the platform. Xie et al. [36] consider two approaches to incorporating quality information in multivariate visualization. Trellis displays [3] combine several visualizations into one panel. Polaris [28] (and Tableau) provides for a more structured analysis process than the DataMeadow.

3 Requirements

This section contains a listing of both functional (task-centered) and non-functional (general) requirements for a visual analytics application designed for multivariate data. These requirements have been derived from treatments on visual exploration [16] and the analysis process [30], as well as the cognitive task analysis in [35].

The primary users of the DataMeadow tool are experts familiar with multidimensional data manipulation and representation. Some of the operations, such as filter and set operations, are too complex for a novice user to easily grasp, yet are necessary to satisfy the requirements of the target user group.

3.1 General

One of the main distinguishing features of visual analytics is the need for powerful and effortless interaction across several visualizations. This goes beyond individual graphical representations—analysts must be able to combine several visualizations in order to correlate findings and insights. Below are the main non-functional requirements of our method necessary to fulfill analyst goals:

(R1) *interaction* — interaction must be smooth and effortless;

(R2) *exploration* — encourage data exploration by providing easy access to analysis tools such as filtering, sorting, correlation, etc [26];

(R3) *iterative refinement* — the approach should lend itself to progressive analysis [11] of the data in small multiples [31]; and

(R4) *communication* — the system should support the production, presentation and dissemination of analytical results [30, 32].

3.2 User Tasks

Visual exploration [16, 19] often follows the "information-seeking mantra" [26]: overview first, zoom and filter, and provide details on demand. Any visual analytics application should support these basic tasks.

More specifically, in this work we are targeting simultaneous visualization of multiple large-scale data sets. The main user task the application needs to support is *comparison*; either comparison between different datasets, such as data for different states in the United States, or between subsets of the same or different sets, such as data for different cities or counties in the same state.

In the task taxonomy of Amar et al. [1], comparison is classified as a higher-level meta-operation. In our model of the DataMeadow,

this is certainly true: in order to support this broad comparison operation, we must provide for a wide range of lower-level user tasks such as (using the terminology of Amar et al.) retrieve value, filter, correlate, characterize distribution, etc. Wehrend and Lewis [33] refer to this operation as compare within and between relations—this also applies to the DataMeadow, where we support both comparison between datasets as well as between subsets within the same dataset.

4 The DataMeadow Method

The DataMeadow method is designed for visual analytics of multiple high-dimensional datasets. The main driving user task behind the design of the technique is comparison between different sets or subsets of data. In this section, we describe the visualization method, including the user tasks supported, the visual mappings, and the interaction techniques.

4.1 DataMeadow

The DataMeadow is an infinite 2D *canvas* and a collection of *visual analysis elements* used for multivariate visual exploration. A visual element is a graphical entity with an appearance, a number of user controls, and input and output *dependencies*. Elements can be created, modified, and destroyed as needed. Individual elements can be chained together using dependencies and then compared to each other. Dependencies and different analysis operations and interaction techniques can also be used to construct more complex visual queries.

More specifically, the DataMeadow consists of the following components:

- **Visual analysis elements.** A graphical entity used for data analysis. Different element types perform different operations. Example types include DataRoses, textual annotations, data viewers, etc.
- **Dependencies.** Directed connections linking one visual element to another. Data cases pass through the dependencies from the source to the destination element.
- **Canvas.** Infinite 2D plane on which all components are anchored. Supports sort and layout operations of elements. Has an associated *data format* that describes the meta information about the available dimensions and their data type.

Each DataMeadow conforms to a specific *data format* that describes the format of the datasets, i.e. the columns and their metadata (column name, data type, etc). The data format specifies what information is stored in the visual elements and is passed through the dependencies connecting them. The meadow can contain several different datasets as long as they all conform to the data format, allowing for comparison of multiple related datasets (such as baseball statistics for different seasons or US Census data from different years).

4.2 Visual Analysis Elements

The basic building block of the DataMeadow method is the visual analysis element, a component consisting of a visual appearance, a variable number of user controls (none for some elements), and input and output dependencies. Each element follows a strict multivariate data model based on the currently active data format for the canvas. This data model governs how information flows through the system through the dependencies and how it can be transformed by the elements.

There are three types of visual elements in the DataMeadow method (examples of each type are given in brackets):

- **Sources.** Producer elements from where data originates and is passed through outgoing dependencies. [database readers, noise generators, number generators, etc]
- **Sinks.** Consumer elements that accept incoming data and consume it, potentially changing its visual appearance to reflect the nature of the data. [viewers, labels, flags]
- **Transformers.** Input/output elements who transform incoming data using some operation and outputs it to outgoing dependencies. [DataRoses]

In the following sections we will be describing some of the elements in greater detail.

4.3 Dependencies

A dependency is a directed connection between two visual elements on the same DataMeadow. This is the basic principle supporting the *iterative refinement* requirement from Section 3.1. Data cases from the source flows along the dependency to the destination element using the data format of the meadow.

Dependencies are never filtered or constrained; all filtering is performed in the visual elements. Mutual or circular dependencies are not possible on the DataMeadow canvas due to the flow-directed nature of the underlying data model.

Dependencies will ensure that changes in source data are properly propagated to all dependees. Thus, when an analyst changes the parameters of a visual element in a chain, all elements further down in the chain are immediately updated to provide feedback to the user. This way, the user can directly see the effect of a parameter change to the visual query.

4.4 DataRose

The core visual elements in the DataMeadow method are called DataRoses: 2D starplots displaying multivariate data of the currently selected dimensions of the dataset. The data can have different visual representations depending on the task; examples include color histogram mode, opacity band mode [10], and standard parallel coordinates mode. The design intention of the DataRose is to provide a self-contained visual entity that lends itself to side-by-side comparison to other datasets.

A DataRose represents one specific dataset, and can be derived either from a database source or be the result of a set operation (see below for more on this). More specifically, a DataRose is a mathematical set, i.e. all entities contained in a rose appear only once.

4.4.1 Visual Representation

Figure 2 shows the three visual rose representations for a fictitious university student database. The database records 500 students and maintains five dimensions: the age (quantitative), major (nominal), gender (nominal), GPA (quantitative), and graduation year (ordinal) of each student. For all three visual representations, a single black polyline is used to show the average for each dimension. Low values are close to the origin, high values reside on the outer radius.

In **color histogram mode**, the data distribution for each dimension is shown on the surface of the rose using a continuous color scale. The color transitions between color values of adjacent axes are rendered using smooth interpolation.

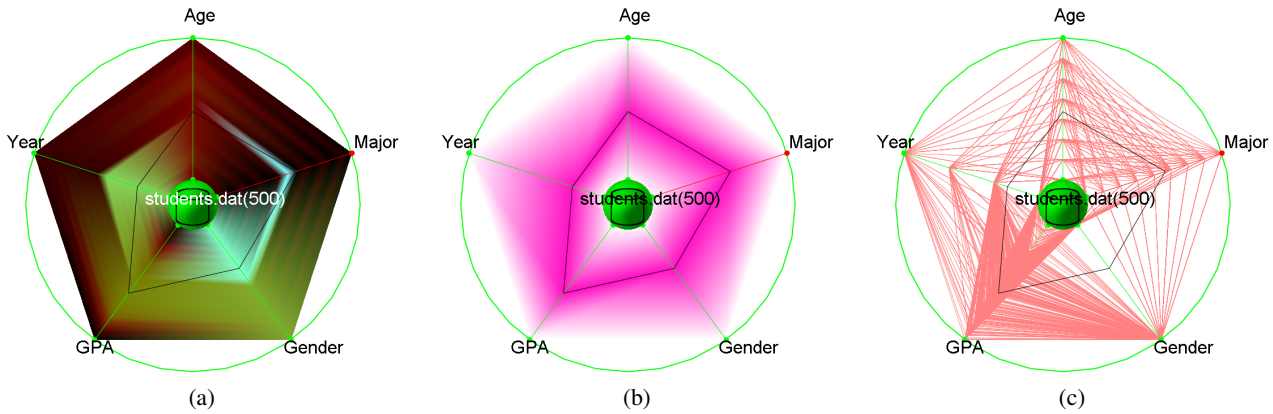

Figure 2: Sample DataRose visualization for a university student database of a computer science department. (a) Color histogram mode (high brightness equals high density). (b) Opacity bands mode. (c) Parallel coordinate mode.

Figure 2a shows a color histogram using the OCS [21] color scale. High brightness indicates high density in the underlying distribution, so it appears that age (the 12 o'clock dimension) is fairly evenly distributed across the dataset. Going clock-wise around the rose, for major there is a concentration of data around the mid-term mark of the dimension. As it turns out, this is a database of students attending courses in a computer science department, and looking at the value legend reveals that this particular value is for students who have computer science as their major. The gender dimension reinforces this fact, as there appears to be a skewed gender balance in the dataset. The students have an above-average GPA, and most of them seem to be freshmen or sophomores.

In **opacity band mode**, the underlying data is abstracted using opacity bands that smoothly go from full opacity at the average to full transparency at the extremes (minima and maxima). Transitions between adjacent axes are again rendered using smooth interpolation. Figure 2b shows an opacity band where the amount of purple color indicates the data density. The same trends we noted from the color histogram representation are visible here as well, albeit at a higher abstraction level. Furthermore, the density of the data for different values is less obvious, and the observation about most students being computer science majors is hard to make here.

Finally, the **parallel coordinate mode** uses traditional parallel coordinate rendering, where all cases of the underlying dataset are rendered using polylines that connect the values for each dimension. However, the downside is that data distribution is more difficult to see in this visual representation.

Accordingly, different representations are suitable for different tasks; while parallel coordinates certainly display the most information, it is sometimes useful to be able to abstract away some of the details when trying to get an overview of the dataset. Opacity bands are suitable for getting an idea of the average and extreme values of the underlying dataset. For some analysis tasks, it is important to be able to see the data distribution, something which can be very difficult in parallel coordinate mode where a lot of data cases might map to the same position on the axis (especially for nominal dimensions). Color histogram mode shows a detailed breakdown of how the data cases divide among the values along each dimension.

4.4.2 Starplot Layout

DataRoses are constructed by splitting a full 360° circle into n parts, one for each of the data fields $F = \{f_1, f_2, \ldots, f_n\}$ to be visualized. This will assign each field $360°/n$ of the circle. For each data field, an axis is drawn radially from the center of the circle to its perimeter. The center part of the rose is reserved for interaction, such as dragging the rose and creating dependencies, and this part is also used for the visual icon for the specific rose type. The remaining part of the axis is normalized to the range of the associated data field and is used for plotting individual data cases.

Note that in all visual modes, we use the starplot axes as continuous dimensions even for nominal data. This is perhaps counter-intuitive and imposes an artificial ordering between these values. For future iterations of the technique, it would be useful to employ the DQC [22] reordering approach to impose an optimal ordering of coordinate mappings of nominal variables.

4.4.3 Axis Filtering

In the DataMeadow, as shown in Figure 1, each DataRose starplot axis also has a dynamic query slider to allow for axis filtering [23]. The handles for each slider are shown as small circles on the axis plotted at the extremes of the current filter selection. In addition, a semi-transparent area is drawn over the areas of the DataRose falling outside of the current filter selection. The user can grab the query handles and move them, dynamically changing the filter selection and causing the visual elements further down in the chain of connected elements to be updated. This allows the analyst to go back and make upstream filter changes that affect a whole query.

This iterative refinement using dynamic queries is an important distinction to software systems that are based on dynamic queries, such as Spotfire. In these systems, the DQ sliders are typically global in scope, whereas they are local for the data flow chain in the DataMeadow.

Figure 3 shows an example of axis filtering where the analyst has filtered the student database example from above to only include students of 25 or above with a certain range of graduation year, major, and GPA. Any outgoing dependencies from this rose will only propagate the filtered data. Furthermore, the data flow model shows interactive feedback at all times as the analyst is changing the dynamic queries, promoting visual exploration of the data.

In Figure 1, the analyst has constructed a complete visual query from a house database for the state of Vermont. By changing the DQ filter settings in the middle rose, the analyst is able to study the correlation of high value and acreage on the number of rooms and bedrooms of a house by looking at the average and extreme values in the result rose to the right. The leftmost database rose and

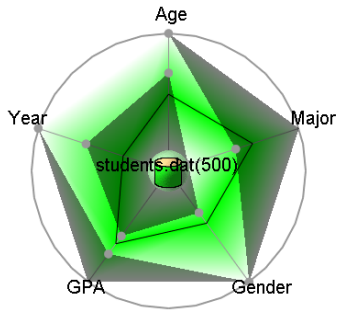

Figure 3: Dynamic query axis filtering for the student database.

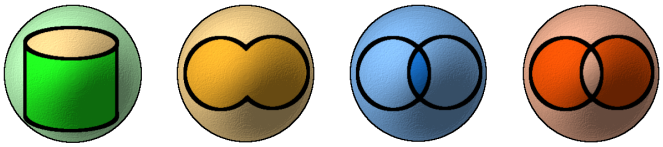

Figure 4: DataRose type icons. (a) Database (source). (b) Union. (c) Intersection. (d) Uniqueness.

the result rose have also been connected to a barchart viewer (see Section 4.5) to show the relative sizes of the two roses.

4.4.4 Rose Types

In order to support complex user tasks such as correlation and characterization, we introduce additional DataRose types other than the standard **source**, which represents an external database loaded from a file. We define the rose types as set operations, allowing us to construct advanced visual queries through constrained and unconstrained dependencies. All rose types accept variable input dependencies, i.e. they have been generalized from standard set theory operations.

- **Source.** External database loaded from a file (see Figure 4a).
- **Union.** Set representing the union of all input dependencies, i.e. the combination of all input cases (see Figure 4b).
- **Intersection.** Set representing the intersection of all input cases, i.e. only cases that are present in all input dependencies (see Figure 4c).
- **Uniqueness.** Set representing unique inputs, i.e. only cases that exist in only one input dependency (see Figure 4d).

Set operation rose types are useful for advanced correlations, such as between different visual query branches. For example, in the case study below, the analyst uses an intersection rose to see whether any of the high value houses he has identified in one visual query also are present in the high acreage subset he derives in another (see Figure 6).

Additional rose types representing other, more complex multi-set operations can easily be added.

4.5 Viewer Elements

Viewers are sinks that accept input and have no output dependencies, typically changing their visual representation to reflect the incoming data. They are useful for studying the results of more complex queries involving DataRoses. The following viewer elements are supported by the DataMeadow canvas:

- **Quantity barchart.** Shows the relative amount of cases coming in from the different dependencies as a barchart.
- **Quantity piechart.** Same as the above, but using a piechart representation.
- **Linear histogram.** Data distribution of each dimension shown as a standard linear histogram.

Examples of viewer elements can be seen in Figures 5 (barchart and piechart).

4.6 Annotation Elements

Annotations are sink elements whose primary purpose is to support the *communication* requirement by providing a way for the analyst to incrementally annotate findings using free-text messages and media. Because they are sinks, an annotation object typically has inbound dependencies, and can thus present reports on the data. The following annotation element types are supported in the DataMeadow:

- **Labels.** Names and labels to denote a specific element or analysis result.
- **Notes.** Longer textual descriptions (more than a single line).
- **Images.** Bitmap images to illustrate particular elements or analysis results.
- **Reports.** Textual reports of the incoming data, such as average, minima and maxima, etc. Automatically updated as the data changes.

4.7 Interaction Techniques

The DATAMEADOW implementation provides a number of interaction techniques (supporting the interaction requirement of Section 3.1):

- **Mouse navigation.** The viewport can be panned by pressing the center mouse button and dragging, or zoomed in or out by pressing the right mouse button and dragging.
- **Brushing.** Selecting a data case in one DataRose will highlight the case in all of its appearances in other DataRoses (parallel coordinates only).
- **Mouse gesture detection.** The user can perform complex mouse gestures on the canvas to create new set operation DataRoses.

The mouse gesture support allows the analyst to easily construct visual queries without having to leave the visualization window to access menu options or even having to use the keyboard. For example, drawing a U-shaped pattern on the canvas will create a union rose, and an upside-down U will create an intersection rose.

4.8 Layout Mechanisms

The DataMeadow canvas lends itself nicely to employing a number of layout mechanisms for arranging the roses and their dependencies. A number of simple layouts such as circle, grid, and dependency depth order are supported. A more complex physically-based layout scheme using springs and dampers can also be employed to provide a more visually interesting and dynamic layout that encourages exploration. The ambition is to provide semi-automatic layout (akin to [35]) to aid the user in organizing the visual elements.

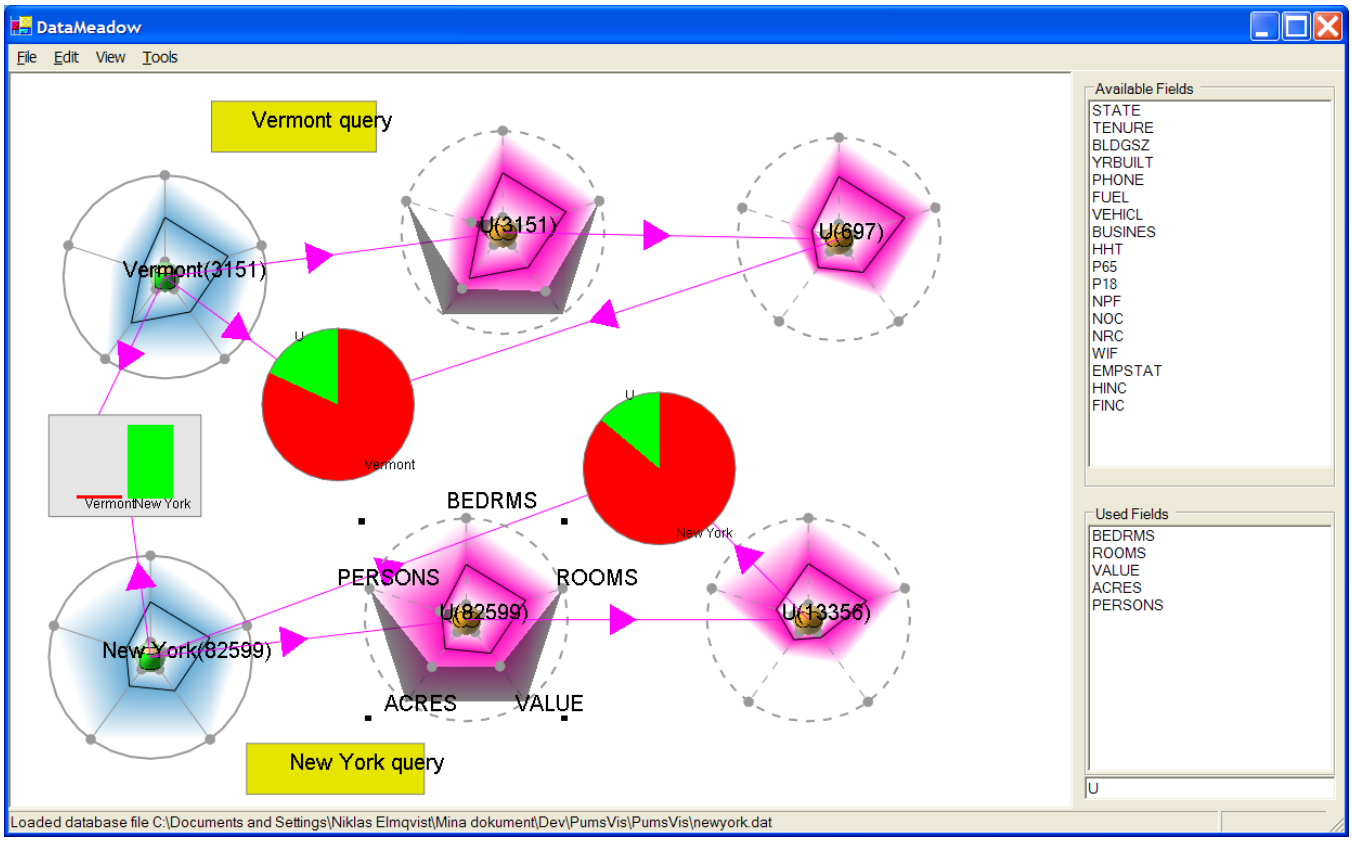

Figure 5: The DataMeadow prototype implementation. The main panel shows the visualization canvas and the smaller panels to the right show the available and currently visualized dimensions in the data.

4.9 Prototype Application

As can be seen from Figure 5, the prototype implementation has three distinct interface parts: (i) a main visualization window, (ii) a dimension selection part (upper right), and (iii) a currently visible dimension part (lower right). The main visualization window is a continuously zoomable viewport into the infinite 2D canvas representing the DataMeadow. Users can easily zoom and pan across the whole canvas using simple mouse interactions. The dimension selection interface boxes allow the user to easily select which dimensions in the data format to visualize—this can be dynamically changed, so that dimensions can be added or hidden as necessary.

4.10 Implementation

The DATAMEADOW application was implemented using the C# programming language and the Microsoft .NET framework. The application uses the Tao bindings for OpenGL to get access to both 2D and 3D accelerated graphics functionality but no special visualization toolkit was used. The interface components were realized using the Windows Forms toolkit.

The prototype implementation has been optimized to deliver interactive framerates even for very large datasets (more than 500,000 data cases). This is primarily possible through the use of the discrete polygon rendering approach for the color histogram and opacity bands modes of the DataRose; parallel coordinate rendering has a much larger performance overhead and is discouraged for datasets of this size (more than 100,000 entities).

5 CASE STUDY: US CENSUS DATA

Let us follow a fictitious analyst (Alan) who is using the DataMeadow to study the Public Use Microdata Sample (PUMS 1%) of the US Census data from 2000. The prototype implementation has support for loading data formats based on either the person or housing records of the PUMS dataset. This allows Alan to easily select and load the database file for a specific state into the application. Alan is interested in studying the PUMS housing records, so he first loads the housing data format. He then decides to start his analysis in the state of Vermont, so he loads this dataset into the application.

Upon finishing loading, Alan is presented with an empty DataRose representing the Vermont dataset, containing 3151 entries. First, he selects which of the 18 dimensions in the database he wants to display, opting for build year, number of rooms, number of bedrooms, acreage, value, and owner income. He quickly creates a data flow chain by right-clicking and dragging on the Vermont rose to create a first derived rose, and then again on the first derived rose to create a second. He will use the first derived rose for filtering, and the second to view the results, so he labels them accordingly. Finally, he creates a barchart viewer and connects the Vermont rose to the result rose so that he can easily observe size ratios as he explores the data. See Figure 1 for his starting setup.

Now Alan is free to get a feeling for the data by changing the filter selection on the filter rose. He does this by clicking and dragging on the DQ handles on this rose and observing the visual results in the results rose as well as the barchart. He is able to quickly confirm some things that he already knows: for instance, that high value and

high acreage implies many rooms and bedrooms.

Next, Alan wants to start a new line of reasoning, so he creates a second two-element chain of derived roses from the Vermont database. He is free to leave his first query undisturbed. He decides to remove the number of bedrooms dimension and instead look at the number of persons in the household. Feeling that he may be on to something, he decides to cross the results of the first query with the results of the second. In order to do so, he creates an intersection rose and connect the two queries to it. This rose will now show the houses from the original dataset that are part of **both** results from the two separate queries. See Figure 6 for the state of his DataMeadow canvas.

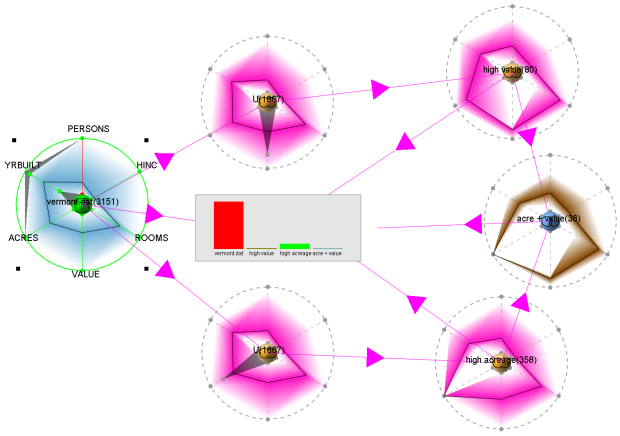

Figure 6: Two visual query branches (value and acreage) crossed using an intersection rose (brown).

Alan decides to bring the state of New York into the picture to contrast against Vermont. He gets rid of the second query branch and loads the New York dataset, resulting in a second blue database rose. All dimension axes are automatically rescaled by the application to use the same scaling factor so that it is possible to directly compare roses from two different datasets against each other. Alan builds up a new query chain for New York and starts exploring the data using axis filtering. By imposing the same constraints on the chains of both states, he can see differences in the datasets. At one point, he notices that in Vermont state, a high number of persons in a household often implies a large acreage, but that this is not at all the case for New York state. See Figure 7 for his final analysis result.

6 User Study

We conducted a qualitative expert review on our prototype implementation. Our goal was to explore the capabilities of the method and gain an idea of its utility. The study involved two visualization researchers from the field. Neither of the two had prior knowledge of the tool.

6.1 Procedure

We structured our expert review based on the US Census 2000 PUMS dataset and a number of questions to drive the visual exploration. In total, there were nine open-ended questions divided into three different groups (inspired by the conceptual levels for situation awareness [9]): direct facts (what is the average house value in Georgia?), comprehension (which state has the highest ratio of small and expensive houses?), and extrapolation (is there a relation

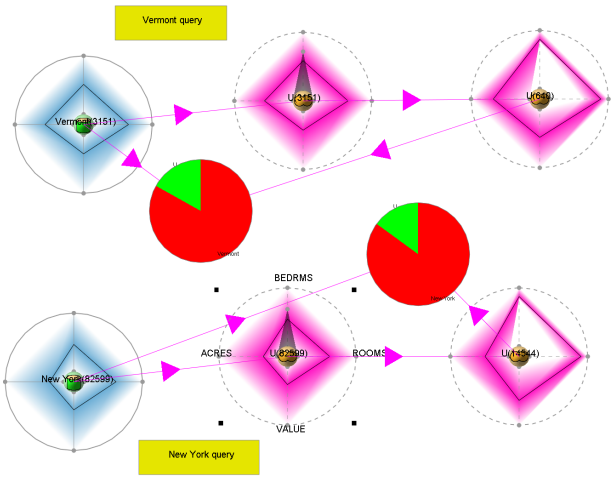

Figure 7: Comparing person data for houses of Vermont (upper branch) and New York (lower branch).

between fuel type and building size in Alaska?). In this way, we hoped to be able to evaluate all aspects of our method.

Our two experts were introduced to the DataMeadow using a tour of the system in which the experimenter showed its main features and analysis methods. This tour lasted ten minutes. After that, the participants were allowed to familiarize themselves with the application. This session typically lasted ten minutes as well.

During the solving of the nine questions on the US Census dataset, the participants were instructed to follow a think-aloud protocol. Only four out of fifty available states in the PUMS dataset were included in the study. Each evaluation session lasted around one hour in total. At the end, we conducted a short free-form interview about their experience using the tool.

6.2 Results

We intentionally designed our nine questions to be of an open-ended nature—we were not interested in quantitatively recording the performance of our experts, but rather to have them exercise all parts of the system and get their feedback on its utility. Still, both participants were able to arrive at answers to all questions.

The participants liked the free-form type of interaction and both remarked it was a good match to how one might think about the analysis process. Being able to filter *in situ* on the dimension axes themselves seemed a good match to how one might think about multidimensional filtering. The ability to "play" with the filter settings at different levels in a dependency chain was often used to both form hypotheses and to inform the next line of reasoning.

Both participants thought that the opacity bands representation was the most efficient for general analysis. In some cases involving the distribution of mostly nominal data (e.g. fuel type), the color histogram was used. Participants remarked that this representation was often too dark because the data was often distributed rather evenly across the dimensions, resulting in only the lower half of most color scales to be used. None of the participants really liked the parallel coordinate representation, remarking that it "showed too much" for the analysis task they were doing.

Some improvements that were pointed out were to include viewers with logarithmic scales to avoid one dataset dwarfing another, to be

able to copy query filter settings from one rose to another, and to be able to set color scales for individual data roses.

7 Conclusions and Future Work

This paper presents a visual analytics method called the DataMeadow for reasoning about multiple large-scale sets of multidimensional data. The primary user task supported by the method is comparison, a high-level meta-task that requires a considerable number of low-level user tasks such as retrieve value, correlation, and filtering. The method consists of an exploratory 2D canvas and individual datasets called DataRoses. DataRoses are variable-dimension starplots that employ a visual multivariate data representations to visualize the data distribution along the coordinate axes being displayed. To summarize, the contributions of this paper are the following:

- a highly interactive canvas (the DataMeadow) for multivariate data analysis;
- a visual representation (the DataRose) based on axis-filtered parallel coordinate starplots that can be linked together to form complex and dynamically-updated visual queries; and
- results from a user study indicating that our method is a useful way to reason about and query multivariate data.

In the future, we expect to integrate additional visual representations into the DataMeadow. Another interesting approach would be the use of both non-standard input devices (e.g. stylii and pen-based interfaces) and output devices (large displays) for the application.

References

[1] 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.

[2] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[3] R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996.

[4] F. Bendix, R. Kosara, and H. Helwig. Parallel sets: A visual analysis of categorical data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 133–140, 2005.

[5] S. E. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. S. Warren, and A. Kaufman. Toward a multi-analyst, collaborative framework for visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, pages 129–136, 2006.

[6] D. Brodbeck and L. Girardin. Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 197–201, 2003.

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

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

[9] M. R. Endsley, B. Bolté, and D. G. Jones. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, 2003.

[10] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the IEEE Conference on Visualization*, pages 43–50, 1999.

[11] D. Gotz, M. X. Zhou, and V. Aggarwal. Interactive visual synthesis of analytic knowledge. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, pages 51–58, 2006.

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

[13] A. Inselberg. Multidimensional detective. In *IEEE Symposium on Information Visualization*, pages 100–107, 1997.

[14] B. Johnson and B. Shneiderman. Tree maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization*, pages 284–291, 1991.

[15] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, Jan./Mar. 2000.

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

[17] D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets? *Information Visualization*, 1(1):20–34, 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] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of the IEEE Conference on Visualization*, pages 230–237, 1990.

[21] H. Levkowitz and G. T. Herman. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, Jan. 1992.

[22] 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.

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

[24] B. Shneiderman. Tree visualization with treemaps: a 2-D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, Jan. 1992.

[25] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, Nov. 1994.

[26] 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, 1996.

[27] M. Sifer. User interfaces for the exploration of hierarchical multidimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, pages 175–182, 2006.

[28] C. Stolte and P. Hanrahan. Polaris: a system for query, analysis and visualization of multi-dimensional relational databases. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 5–14, 2000.

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

[30] J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2005.

[31] E. R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[32] F. B. Viégas and M. Wattenberg. Communication-minded visualization: A call to action. *IBM Systems Journal*, 45(4):801–812, Apr. 2006.

[33] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the IEEE Conference on Visualization*, pages 139–143, 1990.

[34] 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.

[35] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis: Concepts and evaluation. In *Proceedings of the ACM CHI 2006 Conference on Human Factors in Computing Systems*, pages 801–810, 2006.

[36] Z. Xie, S. Huang, M. O. Ward, and E. A. Rundensteiner. Exploratory visualization of multivariate data with variable quality. In *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, pages 183–190, 2006.

[37] J. S. Yi, R. Melton, J. Stasko, and J. Jacko. Dust & Magnet: Multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.