A Scalable Architecture for Visual Data Exploration

Jonathan Decker*
Naval Research Laboratory

Alex Godwin†
Naval Research Laboratory
UNC Charlotte

Mark Livingston‡
Naval Research Laboratory

Denise Royle§
Naval Research Laboratory

ABSTRACT

Intelligence analysts in the areas of defense and homeland security are now faced with the difficult problem of discerning the relevant details amidst massive data stores. We propose a component-based visualization architecture that is built specifically to encourage the flexible exploration of geospatial event databases. The proposed system is designed to deploy on a variety of display layouts, from a single laptop screen to a multi-monitor tiled-display. By utilizing a combination of parallel coordinates, principal components plots, and other data views, analysts may reduce the dimensionality of a dataset to their most salient features. Of particular value to our target applications are understanding correlations between data layers, both within single views and across multiple views. Our proposed system aims to address the limited scalability associated with coordinated multiple views (CMVs) through the implementation of an efficient core application which is meant to be extended by the end-user rather than a programmer.

Keywords: Coordiated Multiple Views, Visual Analytics

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces; E.1 [Data Structures]: Tables; H.4.2 [Information Systems Applications]: Types of Systems—Decision support

1 INTRODUCTION

The amount of data currently available overwhelms our capacity to perform analysis on it. Therefore, building tools to sort through data and determine trends or patterns amongst those variables becomes important first-step to analyzing real-world data. Geospatial analysts must sort through many layers of data to determine which variables may be useful as predictors of adverse events. A method that enables an analyst to assess the risk based on demographic data, terrain information, and any prior events would be a powerful tool in threat analysis. Such a tool would need to combine traditional data analysis techniques with visualization methods in a way that allows a non-expert in statistics and visualization to navigate and interpret the data. While there are many ways to approach such issues, we have chosen to provide multiple views of the data as our first step. Some views use spatial layouts (an important view for reasoning about geospatial information), and some visualize numerical analysis (such as principal component analysis) or help find correlations in the data (e.g. parallel coordinates).

Flexibility within an application of this nature is paramount. Through analysis of the problem domain and interaction with experts in the field, visualization designers may anticipate many of the needs of the end user. However, by providing a means for interacting with existing visualizations as well as specifying new ones, designers provide a more robust toolkit for suiting a wide variety of problems and user preferences. If a user requires a means of analysis that is not currently present in the provided set of tools, then they have the ability to create a new one without resorting to other software packages that are disconnected from the current data suite and possibly unfamiliar to the user. Flexibility also provides a greater preference for adjusting to user preferences in mapping of visual objects to user tastes as well as to their particular strengths and interests.

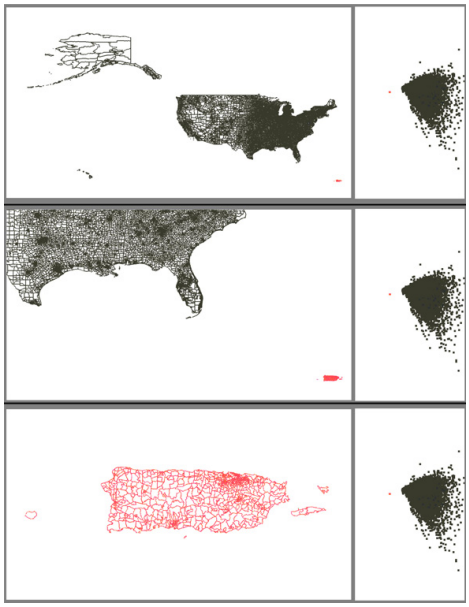

Figure 1: An example of using multiple views to make a simple discovery. Here an outlier in the PCA scatter plots is highlighted on the right, and this turns out to be all the tracts of Puerto Rico, which have undefined values for almost all the variables used. Finding bad data is important, since it can throw off analysis.

The focus of this paper is the flexible exploration of data through the creation of coordinated multiple views (CMVs). Our work extends design patterns presented by Weaver in Improvise [5] and Heer in prefuse [4]. However, existing CMV applications are limited in their ability to manage interactive performance when viewing large datasets over 100000 records, and have little to no support for dynamic or complex data, faults that have been outlined by Andrienko and Andrienko [1]. We feel that the issue of scalability is a major problem with existing CMV research. The only solution is a complete reversal in the paradigms presented out by previous research, primarily the concept of deep class hierarchies and easy of development through managed language and modular Object-Oriented Programming (OOP). These practices incur overhead which can significantly reduce the capacity of an application to read, interpret and display data. Instead, the source code should be written to be as efficient as possible, and the burden of extension should be shifted from the programmer to the end-user through scripting languages and user-created views based on a set of predefined data operations and visual metaphors.

*e-mail:decker@itd.nrl.navy.mil
†Now at Charles River Analytics e-mail:agodwin@cra.com
‡e-mail:mark.livingston@nr.navy.mil
§e-mail:denise.royle@nr.navy.mil

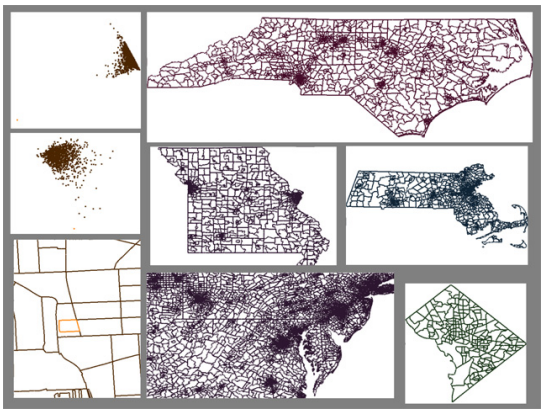

Figure 2: Clockwise from bottom center: Zoomed view of all census tracts, zoomed view of MD tract, two PCA plots of MD, NC tracts, MA tracts, and DC tracts. The center view displays the MO tracts.

2 Related Work

Flexibility and scalability in the analysis of incoming data are crucial in the arena of Defense. Experienced analysts often have a routine developed for the fast fusion and interpretation of incoming data, and a system that is limited in its adaptability to the needs of an analyst will go unused. Component based visualization is frequently accomplished through the use of layout toolkits, as in the Improvise system [5], or as extensible frameworks such as prefuse [4] and the Infovis Toolkit [2].

Our system is designed following several of the design patterns outlined by Heer [3], which are themselves derived from Gamma et. al.'s earlier work. Specifically, we implement the data column, cascaded tables, proxy tuple, and expression patterns in our data model. Unlike systems like Improvise and prefuse that use their own expression language, our expressions leverage the Lua scripting language. Moreover, our system is designed from the ground up for memory efficiency so as to perform interactively on the massive, complex data sets that GIS analysts must work with.

2.1 Expression-Driven Visualizations

Previous guiding principles have frequently expressed the need for an expressive specification of visualization parameters. Heer's design patterns provide an excellent set of guidelines for creating a visualization-oriented data model that supports the use of expressions, and we have incorporated many of these patterns into our own architecture specification [3]. User-created expressions provide an additional layer of customization in visual analysis, particularly in the design of CMV applications [5].

We use the popular column-oriented data model, which allows for dynamic queries for many interactive techniques, such as brushing and highlighting, which is currently implemented in our system (See Figure 1). We uses a forest structure of cascading tables, where each child table sees the columns of its parent as well as its own additional columns. There is also a special type of column that uses a user-defined expression and a set of columns as input parameters. An expression column can be used as a decorating column by simply evaluating a set of the columns in the table model to a color, visibility parameter, size, shape, or other visual property. Expression chaining can be used to great effect by modifying the view parameters and the creation of entirely new visualizations.

3 System Design

The proposed system runs one a single machine with multiple display adapters. Our system takes over monitors and displays a visualization layout predefined in an input file. Each visualization can be set to span multiple displays, and measurements can be provided at initialization to account for the space between monitors and monitor bezels. Views can also be resized and dragged anywhere within the multimonitor space. Data is stored in tables where each column represents the value of every entry for a specific type. The user may specify which datasets are loaded, which tables they are to be stored in, and what additional columns should be defined. Furthermore, cascaded tables can be defined on top of the base tables, which contain a subset of the base table's data, as defined by a filter (See Figure 2).

4 Preliminary Results

We evaluated our system by loading the 2000 Census Tract dataset, which has over 65000 records and over 40 variables. This includes the census tracts themselves, which are stored as rings with 100s of vertices. On a six-tile display wall, we have been able to brush between windows at interactive rates (a few seconds for a complete update between all views). Given that the data is completely loaded into local system memory, the performance is troubling, and does much to solidify the importance of efficient core code.

5 Conclusion and Future Work

The system we have proposed is the foundation for a truly scalable architecture which will allow for visual exploration of large, high-dimensional, complex, and dynamic datasets. This contribution involves a low-level implementation focused on memory and rendering efficiency.

We plan to continue our work by creating new components for the analysis and reduction of the dimensional space of data, such as the Kriging methods which allows analysts the ability to interpolate data values and fill in the missing details. We are currently extending our base application so that full tables can be handed over to Lua expressions.

Additionally, we will incorporate direct dataset access into our column-based table structure, to allow for access to large-scale and/or dynamic data. We will also seek to build not interactive elements into the system, as well as automation based on user-preference and analysis. Finally we will extend our system so that it may run on a computing cluster, which is a crucial step to achieve an optimally scalable system.

Our driving application is the display of GIS data for analysts. We plan to evaluate our system for use in predictive analytics and as a teaching tool for new analysts. The system will undergo usability testing and user evaluation with a subject matter expert in the near future.

References

[1] G. Andrienko and N. Andrienko. Coordinated Multiple Views: a Critical View. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 72–74. IEEE Computer Society Washington, DC, USA, 2007.

[2] J. Fekete. The infovis toolkit. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pages 167–174.

[3] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):853, 2006.

[4] J. Heer, S. Card, and J. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430. ACM New York, NY, USA, 2005.

[5] C. Weaver. Building highly-coordinated visualizations in Improvise. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pages 159–166.