Improvisational Geovisualization of the 2000 United States Census

Chris Weaver

GeoVISTA Center
Department of Geography
The Pennsylvania State University
cweaver@psu.edu

Abstract

The development of software tools for visual data analysis is a recent focus of research in both information and geographic visualization. The goal is to allow users to explore information that contains geospatial, temporal, and abstract components in a flexible, integrated, interactive graphical environment that requires minimal training to use. Improvise is an exploratory visualization software package in which users construct and explore multiple interactively connected views of their data. Improvise gives users precise control over how spatial navigation and item selection affects the presentation of space, time, and abstract data attributes in and between multi-layer maps, scatter plots, parallel coordinate plots, tables, grids, and other views. This paper describes application of Improvise visualization to county-level demographic data from the 2000 United States Census. Tight integration of building and browsing made it possible to realize a usable and useful dynamic display of Census data in a few days. The visualization can be rapidly modified and extended during ongoing exploration and analysis.

1. Introduction

Improvise (Weaver 2004) is an exploratory visualization environment in which users rapidly construct and explore multiple interactively connected views of data. It consists of a graphic user interface on top of a modular library of visualization components, forming a fully implemented, web-capable Java application that appears and behaves like other "office productivity" applications. Users browse visualizations using mouse and keyboard input gestures to navigate in space and select data items. Although interaction is confined to a single view at any given time, tightly coupled coordination between views produces the illusion of browsing each visualization interface as a coherent whole. Improvise has been used to build a wide variety of visualizations, each of which is a self-contained Extensible Markup Language (XML) document that can be saved, opened, copied, and shared as a regular file. Improvise is currently being used for exploration of geographic data from a variety of knowledge domains, including nineteenth century railroad and hotel travel patterns, ham radio communications, Michigan hydrographic layers, and county-level state and federal election results.

In the Improvise user interface, users create, layout, parameterize, and coordinate views using implementations of the *Live Properties* coordination model and the *Coordinated Queries* visual abstraction language. Building occurs inside the same top-level window that contains views. Moreover, building is interactive; changes take effect immediately without the need for a separate compilation stage. This live, amodal interface design allows users to switch rapidly between building and browsing. The goal is to enable exploration that is free form and open-ended, particularly during initial inspection of newly encountered data sets. In other words, the goal is to support *improvisational visualization*.

This paper describes the results of interactive construction and initial exploration of an Improvise visualization of data obtained from the National Atlas (http://www.nationalatlas.gov/). Starting with county-level demographic data from the 2000 United States Census, the visualization was grown through an iterative process of creating views, populating them with data, and coordinating them in terms of user interaction. This tightly integrated method of building and browsing Improvise visualizations made it possible to realize a highly useful and usable dynamic display of Census data in a few days, utilizing multi-layer overview and detail maps, drill-down from states to counties to cities and airports by successive selection and filtering, and bivariate coloring for relating arbitrary user-specified functions of demographic variables. Most importantly, the entire visualization can be rapidly modified and extended to develop hypotheses and exploit discoveries during ongoing geovisual exploration and analysis.

2. Background and Previous Work

Research on *information visualization* focuses on the display of predominately abstract data by *visually encoding* attributes into spatial and other perceptual dimensions such as location, size, shape, color, rotation, and texture (Mackinlay 1986). Information visualizations are composed of *views* that display data and allow user interaction on that data or the space in which it is shown. The display and interaction characteristics of views constrain how data can be visually encoded inside them. Whereas scatter plots typically allow panning and zooming over a 2-D space in which items are graphically embedded using location, size, and shape, table views usually allow vertical scrolling and selection of rows drawn with color, text, and icons.

Information visualizations have several qualities that make them effective for data exploration. First, they can display high-dimensional, abstract data in readily understandable ways. Exploring such data is quicker and easier when it is presented in graphical rather than textual form. Second, they can display data from multiple perspectives simultaneously using multiple views. Different views can show different aspects of the data, allowing users to explore the data in alternate ways at a glance. Third, they are usually highly interactive, allowing users to rapidly select which data items to show and how to show them, using the mouse and keyboard to directly manipulate views. Finally, they can interactively couple the appearance and behavior of multiple views, allowing users to manipulate visualizations as a whole by interacting with individual views.

In visualizations that contain multiple views, interaction can be defined in terms of *coordinations* that determine how the appearance and behavior of each view depends on navigation and selection in other views. Common patterns of coordination include:

- *Synchronized Scrolling*. Two views show the same items from the same data set, the corresponding items in different data sets, or the same region of a coordinate space.
- *Overview + detail*. One view shows details about items selected in another view, such as when a rubber band in one view is coupled with the bounding box of another view.
- *Brushing*. Items selected in a view are highlighted in another view. Highlighting can take the form of any visual differentiation, including invisibility (for visual filtering).
- *Drill-Down*. Selection of an item in a primary view specifies data to be shown in secondary views. Drill-down often involves either accessing a collection of related data sets or aggregation/grouping/categorization of one large data set.
- *Semantic Zoom*. Zooming in a view changes the appearance of items shown in it. Multiple levels of zoom can show progressively higher detail as the user zooms in.

Improvise users create views, access data, specify queries, define visual encodings, and establish coordinations on the fly during exploration and analysis. Although many visualization systems—such as LinkWinds (Jacobson 1994), Tioga-2 (Woodruff 1995), and DEVise (Livny 1997)—possess some of these features, few systems do so in a self-contained, fully interactive visualization construction user interface.

Snap-Together Visualization (North 2000) uses a relational data model that coordinates views using *primary key actions*. When two views are coordinated, invoking an action in one view causes the other view to perform its corresponding action. Actions are extensible and include loading (of data), selection (of items), and scrolling (over a list of items). Unlike Snap, Improvise allows fine-grain user customization of dependencies that involve visual encodings as well as data.

GeoVISTA *Studio* (Takatsuka 2002) is an integrated visualization environment for building geovisualizations interactively using a graph-based visual coordination editor. Any component that conforms to the JavaBeans specification can be a view. Development of new views by the community of GeoVISTA *Studio* users has resulting in a large library of views utilized in numerous visualizations. Improvise and GeoVISTA *Studio* have opposite strengths. Whereas GeoVISTA *Studio* has extensive functionality for representing and displaying geospatial information—based on the GeoTools open source Java GIS toolkit (Liu 2005)—coordinations can be hard to incorporate into its visualizations. Conversely, Improvise has modest geospatial data handling capability but allows interactive construction of a rich variety of coordinations.

3. Improvise

3.1 Coordination Model and Visual Abstraction Language

Improvise visualizations are specified using a visual abstraction language called *Coordinated Queries* that is built on top of a coordination model called *Live Properties*. Live Properties provides a simple, elegant basis for coordinating multiple views through shared interactive parameters that determine what, how, and where views display data. Two *controls* coordinate whenever they are connected through their *properties* via at least one *variable* (Figure 1*)*. Because more than two controls can share the same variable, it is not necessary to connect views pairwise to transitively coordinate sets of views, unlike systems like DEVise and Snap.

Coordinated Queries is an expressive, high-level visualization query language for coordinating access, processing, and rendering of multiple data sets across multiple views. Query expressions specify how to map data attributes into graphical attributes in views. Multiple data sets can be loaded, indexed, grouped, filtered, sorted, and visually encoded in terms of selection and navigation between multiple views. The combination of Live Properties and Coordinated Queries enables open-ended visual analysis by allowing users to design, construct, explore, and extend highly coordinated visualizations of multiple simultaneous data sets interactively.

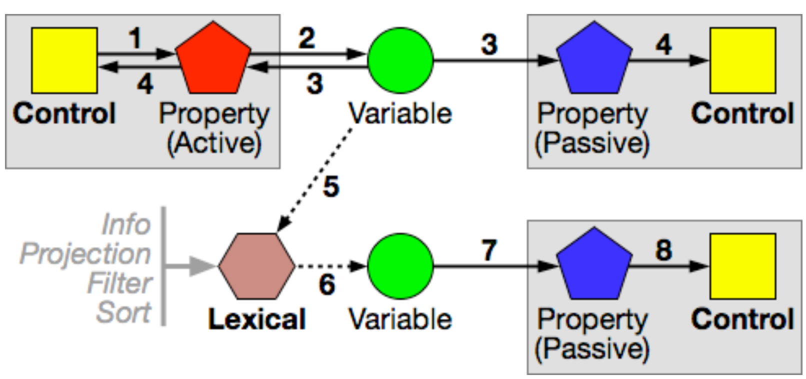

Figure 1: Improvise coordination. In response to interaction, a control modifies the value of one of its properties (1), which assigns the new value to its variable (2). The variable sends a notification to all properties bound to it (3), each of which notifies its control of the change (4). The variable also notifies all *lexicals* (query operations) whose expressions refer to it (5). Each lexical notifies any variables to which it is assigned as a value (6). When a view receives notification through one of its properties that a lexical has changed (7, 8), it updates itself by processing the modified query.

3.2 Software Architecture

The Improvise software architecture (Figure 2) consists of an integrated building and browsing user interface on top of an extensible visualization component library that is modular in terms of:

- *Views*. Displays include maps, scatter plots, time series, 3-D scenes, tables, grids, etc.
- *Glyphs*. Items are visually encoded in views using simple geometric shapes, text labels, arrows, icons, etc. that have position, size, shape, color, and other perceptual parameters.
- *Data Access*. Data sources include local files, compressed archives, web resources, etc. Data formats include delimited text, XML, DBF, Shapefiles, etc.
- *Queries*. Data processing operations include visual encoding, filtering, ordering, aggregation, categorization, indexing, etc.
- *Data Types*. The values of variables, properties, and data items include numbers, strings, booleans, ranges, dates, data sets, lexical query statements, etc.
- *Operators*. Calculations include object construction, type casting, object member access, arithmetic, and statistics, as well as glyph construction and query operations on data sets.

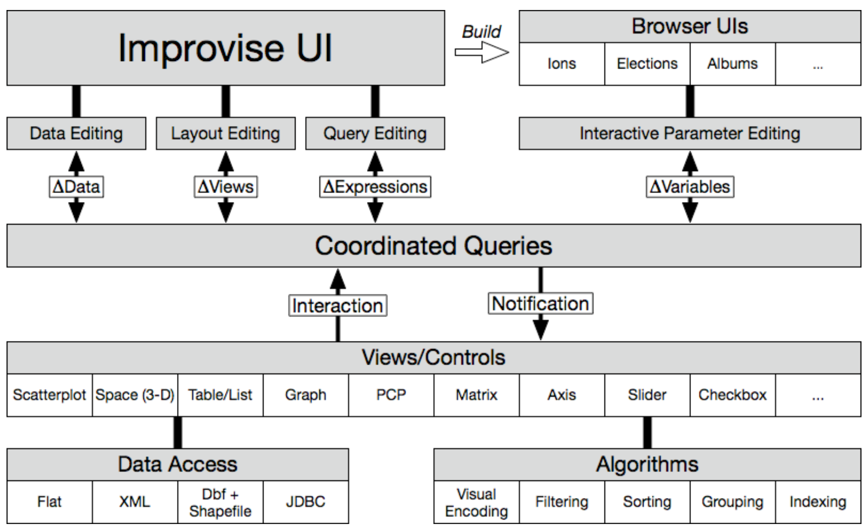

Figure 2: The Improvise software architecture.

Improvise provides a wide variety of view types:

- *Plane views* display scatter plots, time series, histograms, etc. Navigation involves panning and zooming of two ranges.
- *Space views* display 3-D scatter plots, OpenGL scenes, maps with altitude, etc. Navigation involves translating, scaling, and rotating a camera relative to the origin.
- *Lists and tables* display data attributes in one or more columns. Cells contain formatted text and/or icons. Navigation involves standard scrolling.
- *Graph views* display node-and-edge graphs. Packs visually cluster related nodes. Navigation involves dragging individual elements or the entire graph.
- *Grid views* display a 2-D grid of cells. Navigation involves panning over the grid.
- *Ortho views* are individual sections of a parallel coordinate plot (PCP) (Inselberg 1985). Multiple ortho views can be laid out horizontally to form whole PCPs.

- *Arc diagrams* (Wattenberg 2002) connect points along a line using half-circular arcs above or below a divider line, such as for visualizing temporal patterns.
- *Pie, bar, and strip views* display the corresponding charts.

Although Improvise currently draw maps using plane views without geographic projection, near-term plans involve reimplementation with GeoTools. New views under development include zoomable node-and-edge graphs for displaying concept maps (Gahegan 2003) and a technique called *Reruns* that displays overlapping artificial and natural temporal cycles in a wrapping grid view. Improvise also provides common user interface widgets, including checkboxes, labels, text fields, scrollbars, sliders, and scatter plot axes. Instead of displaying data, these controls provide ways to manipulate interactive parameters during browsing.

3.3 Visualization Construction

Improvise follows the multiple document model in which one or more visualizations run in parallel, each in its own top-level window. The builder interface is built around a set of editor dialogs, local to each visualization window, in which designers access data, build expressions for querying data, and create, coordinate, and layout views. The browser interface consists of the set of views as they are laid out inside each window. All changes made in editing dialogs take place immediately and reversibly, resulting in an incremental, organic style of visual exploration.

The expressions that make up query operations are constructed in a tree-based editor. The user builds each expression top-down, by choosing an operator for each position in the tree. Editing this way takes a little getting used to, but has the advantage of being syntactically constrained. Editing is live; the visualization reflects changes immediately. Cloning entire projections, filters, and sorts allows users to experiment with variations of expressions quickly and reversibly. Large libraries can be built up for reuse or rapid switching during visualization design and exploration. The Improvise user interface has several enhancements designed to facilitate visualization construction and exploration, including predefined macros, built-in queries, screenshots, and *integrated metavisualization* that allows users to visualize the coordination structure of Improvise visualizations *in situ* (Weaver 2005). The expression editor incorporates several improvements over standard tree interfaces, including syntactic coloring and full cut-copy-paste and drag-and-drop of subexpressions between multiple editors.

4. The Census Visualization

The Improvise Census visualization is the result of an iterative process of design, construction, and exploration involving three major versions and dozens of minor versions over the course of a year. Evolution of the visualization has been driven primarily by the incremental addition of new cartographic features to Improvise. The visualization displays county-level demographic data from the 2000 United States Census in a style intended to imitate the basic appearance of the paper Census atlas produced by Brewer and Suchan (Brewer 2001), but extended to include views for exploring urban and transportation relationships in more detail.

Displayed data includes Shapefiles and DBF database files, obtained from the National Atlas (http://www.nationalatlas.gov/atlasftp.html), that describes states, counties, cities, urban areas, roads, railroads, and airports. Demographics for each county include population, population density, gender percentages and ratios, median age, percent change since 1990, and proportions of major racial/ethnic groups and age categories.

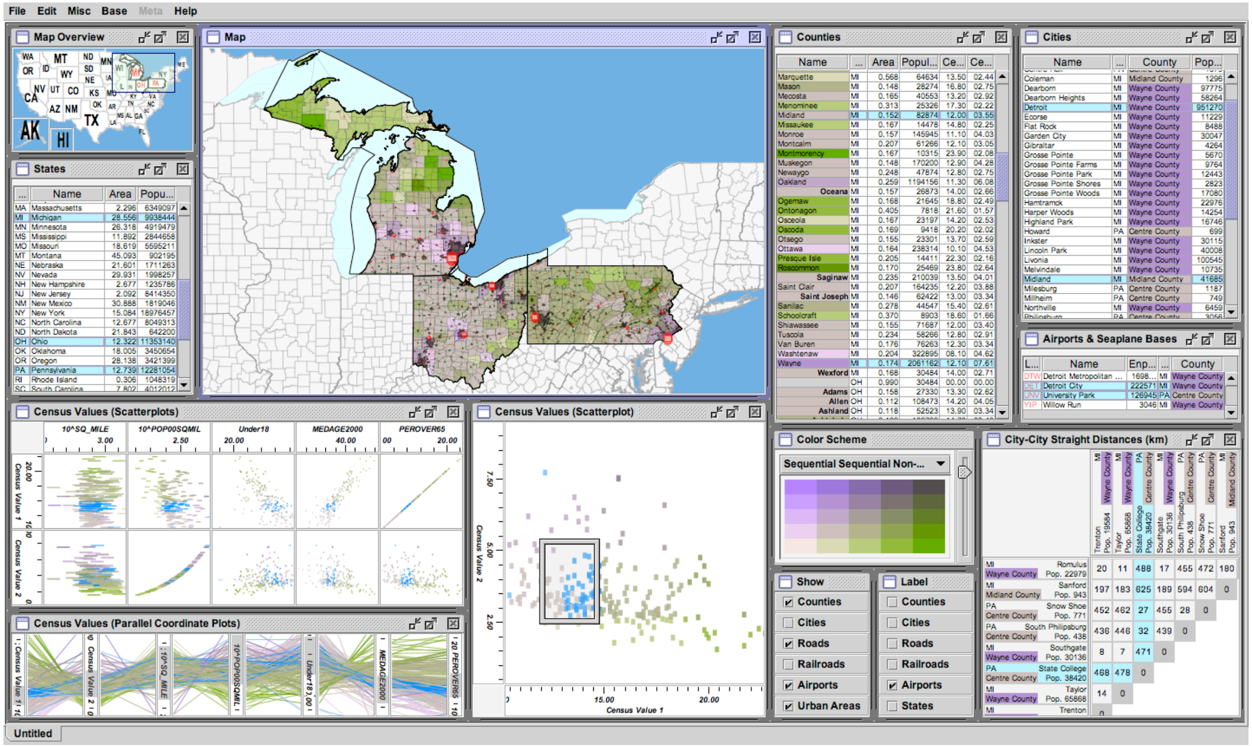

Figure 3: The Census visualization.

The visualization (Figure 3) displays overview and detail maps alongside a list of states and territories. The detail map draws and labels various natural and artificial areas of selected states, as determined by checkboxes. The color used to fill each county is calculated by mapping five equal intervals of two county-level demographic attributes—in this case, population density and population percentage over 65—into a green-purple sequential-sequential bivariate color scheme generated by ColorBrewer (Brewer 2003). Improvements to rendering performance have resulted in navigation and selection coordination that is much faster than in earlier versions, enabling smooth interaction throughout. In particular, the main map contains three *nested lenses* that can be dragged and stretched to reveal features in specific areas of the map (Figure 4). To explore (Figures 5, 6), users can change the color scheme or edit visual encodings and demographic functions by modifying expressions in the query editor (Figure 7).

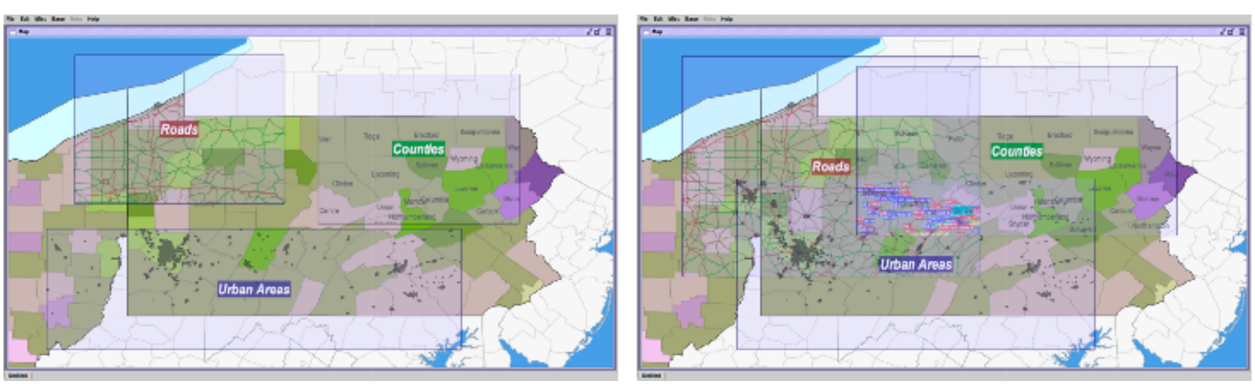

Figure 4: Lenses render additional map layers in translucent frames. Left: Nested lenses draw county names, urban areas, and roads color-coded by type on top of counties. Right: The map labels roads inside the intersection of the lenses.

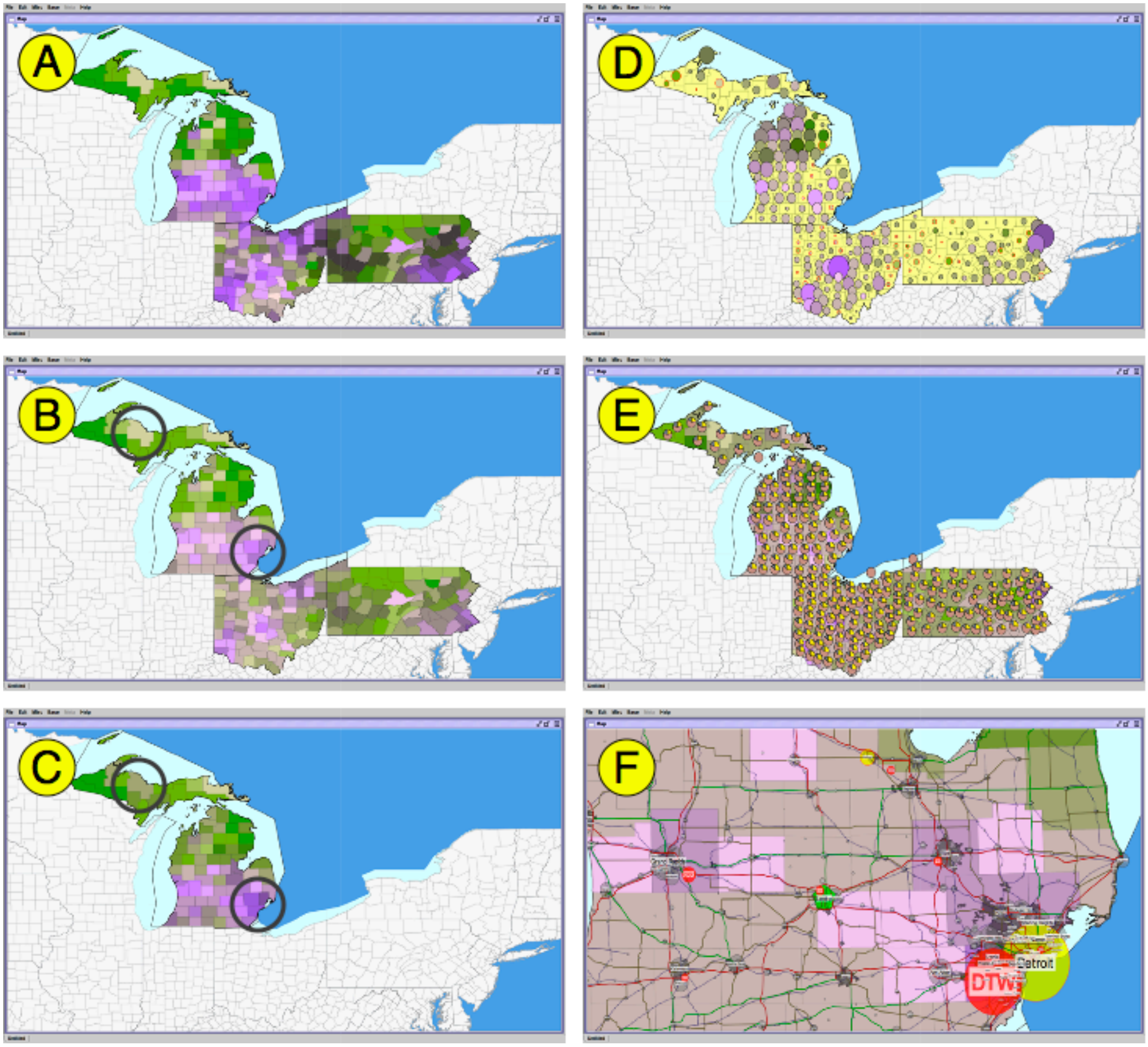

Figure 5: Variations of the main map view, all using bivariate coloring of percent over 65 (purples) against population density (greens). (A) Quantiles. (B) Natural breaks (Jenks). (C) Natural breaks (Jenks) limited to Michigan counties (two differences are circled). (D) Percent population since 1990 as oval size, with losses edged in red. (E) Color-coded pie charts of relative population under 18 (gold), 18-65 (copper), and over 65 (steel). (F) Transportation and city layers turned on.

The other views of the visualization allow comparison of the two demographic functions with up to five demographic variables, using parallel coordinate plots and a scatter plot matrix. One table view lists counties of selected states, using the same color scheme as the detail map. Two more table views list the cities and airports in selected counties. Users can drag and stretch portals (shown as translucent rectangles) in the parallel coordinate axes to highlight a subset of counties in all views. In tables, counties for which all demographic values fall within the corresponding portal ranges are highlighted using right-justified, boldface text. In plots, counties inside all portal ranges are highlighted in cyan. All views support additive selection of multiple data items; the map allows selection of counties by clicking, rubber banding, or lassoing regions. As a result, it is possible to drill-down to look for demographic patterns that involve arbitrary states, counties, cities, and even airports.

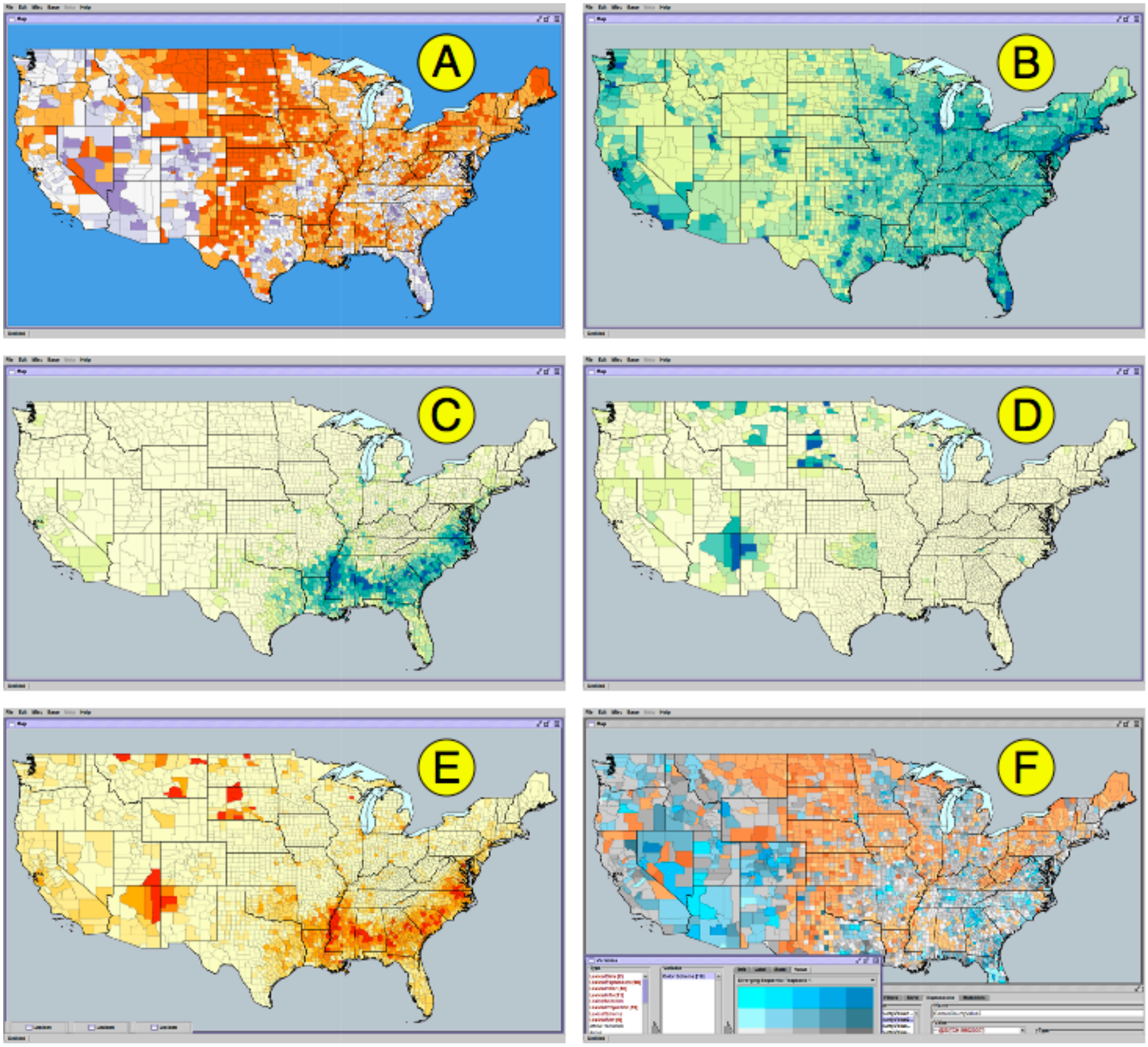

Figure 6: Changing color schemes and demographic functions interactively. (A) Percent change since 1990. (B) Population density. (C) Percent of population, Black or African American. (D) Percent of population, American Indian and Alaska Native. (E) Percent of population, all non-white. (F) Choosing a divergent-sequential bivariate color scheme and editing demographic functions to show male to female ratio against population change since 1990.

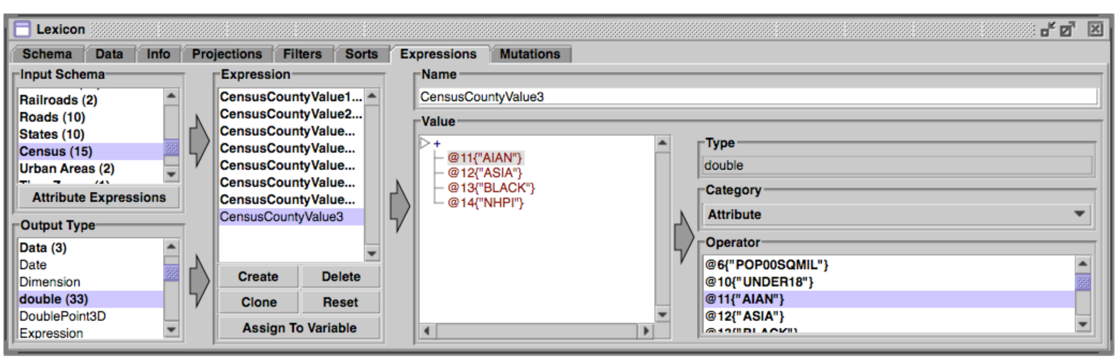

Figure 7: Editing expressions. A derived data attribute is specified as the sum of the percentage of population for four minority groups. In figure 6E, Jenks natural breaks is used to categorize counties in terms of the derived attribute.

5. Conclusion

Improvise consists of a unified builder and browser user interface on top of a modular library of visualization components. In Improvise, visualization designers rapidly create, lay out, and coordinate multiple views of data accessed from one or more flexibly specified sources. Visualization users explore data by browsing the resulting visualization interfaces. Because browsing occurs directly within the builder interface, users with sufficient motivation and skill can act as their own designers, making possible open-ended exploration and analysis. Although Improvise was not designed for mapping applications, ongoing incorporation of cartographic functionality has made it possible to built a usable and useful visualization of Census data. Geographic projection and legends are two features that are planned for addition to Improvise in the immediate future. Improvise is available (with the example visualization from this paper) under the Gnu Public License (GPL) at http://www.personal.psu.edu/cew15/improvise/.

Acknowledgements

Thanks to Alan MacEachren, Mark Gahegan, Donna Peuquet, and the rest of the team at the GeoVISTA Center. Special thanks to Cindy Brewer for offering constant encouragement.

References

Brewer, C. and Suchan, T. U.S. Census Bureau, Census Special Reports, Series CENSR/01-1, Mapping Census 2000: The geography of U.S. diversity.

Brewer, C., Hatchard, G., and Harrower, M. 2003. ColorBrewer in print: a catalog of color schemes for maps. Cartography and Geographic Information Science. 30(5):5-32.

Gahegan, M. et. al. 2003. From concepts to data and back again: connecting mental spaces with data and analysis methods. Proceedings of GeoComputation. Southampton, UK.

Inselberg, A. 1985. The Plane With Parallel Coordinates. *The Visual Computer*. 69-91.

Jacobson, A., Belkin, A., and Orton, M. 1994. LinkWinds: interactive scientific data analysis and visualization. *Communications of the ACM.* 37(4):43-52.

Liu, Y., Gahegan, M., & Macgill, J. 2005. Increasing geocomputational interoperability: towards a standard geocomputation API. Proceedings of GeoComputation. Ann Arbor, MI.

Livny, M. et. al. 1997. DEVise: integrated querying and visualization of large datasets. Proceedings of SIGMOD. Tucson, AZ. 301-312.

Mackinlay, J. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics.* 5(2):110-141.

North, C. 2000. A user interface for coordinating visualizations based on relational schemata: Snap-Together Visualization. PhD Dissertation, University of Maryland.

Takatsuka, M. and Gahegan, M. 2002. GeoVISTA *Studio*: A codeless visual programming environment for geoscientific data analysis and visualization. *Computational Geoscience*, 28(10):1131-1144.

Wattenberg, M. 2002. Arc diagrams: visualizing structure in strings. Proceedings of the IEEE Symposium on Information Visualization 2002. Boston, MA. 110-116.

Weaver, C. 2004. Building highly-coordinated visualizations in Improvise. Proceedings of the IEEE Symposium on Information Visualization 2004, Austin, TX. 159-166.

Weaver, C. 2005. Visualizing coordination *in situ*. Proceedings of the IEEE Symposium on Information Visualization 2005. Minneapolis, MN. 165-172.

Woodruff, A. et. al. 1995. Navigation and coordination primitives for multidimensional visual browsers. Proceedings of Visual Database Systems. Lausanne, Switzerland. 360-371.