Multidimensional Visual Analysis Using Cross-Filtered Views

Chris Weaver*

The GeoVISTA Center and Department of Geography
The Pennsylvania State University

ABSTRACT

Analysis of multidimensional data often requires careful examination of relationships across dimensions. Coordinated multiple view approaches have become commonplace in visual analysis tools because they directly support expression of complex multidimensional queries using simple interactions. However, generating such tools remains difficult because of the need to map domain-specific data structures and semantics into the idiosyncratic combinations of interdependent data and visual abstractions needed to reveal particular patterns and distributions in cross-dimensional relationships.

This paper describes: (1) a method for interactively expressing sequences of multidimensional set queries by cross-filtering data values across pairs of views, and (2) design strategies for constructing coordinated multiple view interfaces for cross-filtered visual analysis of multidimensional data sets. Using examples of cross-filtered visualizations of data from several different domains, we describe how cross-filtering can be modularized and reused across designs, flexibly customized with respect to data types across multiple dimensions, and incorporated into more wide-ranging multiple view designs. The demonstrated analytic utility of these examples suggest that cross-filtering is a suitable design pattern for instantiation in a wide variety of visual analysis tools.

Index Terms: D.2.2 [Software Engineering]: Design Tools and Techniques—[User Interfaces]; H.2.3 [Information Systems]: Database Management—[Languages]; H.5.2 [Information Systems]: Information Interfaces and Presentation—[User Interfaces]

1 INTRODUCTION

Engendering support for analytic reasoning through the development of new visual representations and interaction techniques is a key effort in the research agenda for visual analytics [22]. Many visual data analysis tools, from prototype to production, have been developed to demonstrate such representations and techniques, often with application to and evaluation of data analysis needs in important information domains. Descriptions of these tools nearly always focus either on a single new representation or technique, or on the apparent concomitant applicability of the tool as a whole. They rarely consider underlying patterns of composition of such representations and techniques, both new and long understood, as foundations of analytic utility and usability in visualization design. This is surprising, given the increasingly common—but persistently ad hoc—utilization of coordinated multiple view approaches as scaffolding for compositional design of visual data analysis tools.

Coordinated multiple view approaches are effective for visual data analysis precisely because they support expression of useful multidimensional queries through interaction. Moreover, sustained research in information visualization has identified individual forms of coordination that are particularly flexible and intuitive for expressing certain fundamental visual queries, such as overview+detail and brushing—so much so that particular coordinations and queries are often synonymous. What is largely missing is understanding—let alone formalization—of how particular patterns for composing views and coordinations can be used in the design of tools that support expression of the variegated visual queries needed for far-reaching visual data analysis.

This paper describes one such pattern, *cross-filtered views*, that provides interactive drill-down into interdimensional relationships buried in attribute values spread across one or more data sets. Understanding the general *structure* of a prototypical cross-filtered design is straightforward because it consists of well-known visualization components: (1) multiple coordinated views each support selection over the set of unique attribute values in a data column; (2) each data column is paired with a dimensionally-appropriate type of view that supports indication of attribute values by selection or navigation, such as clicking on dates in a calendar view or rubberbanding regions of values in a scatter plot; (3) users can rapidly toggle brushing filters between pairs of views to pose complex drill-down set queries, even across multiple tables.

Absent from this description is an appreciation for the *process* by which trained domain experts can follow complex lines of inquiry using sequences of simple interactions in the performance of a wide range of general and specific visual analysis tasks: (1) compare values in a view to expose potential relationships between the people, places, and times represented by the data; (2) select values to express a hypothesis that a relationship exists between them; (3) cross-filter other views on those values to explore further within the context of that hypothesis. Repeating these steps enables expression and exploration of more nuanced hypotheses. Moreover, selections are mutable and cross-filters are reversible for any dimension at any point in the process. Hypotheses can be restated quickly and flexibly by adding, removing, changing, and reordering dimensional clauses, all performed by clicking data items or checkboxes. It is not individual views or coordinations but rather their particular and deliberate (but by no means unique) composition that makes such sophisticated visual exploration and analysis possible.

We start by describing an example in which cross-filtering is applied to visual analysis of political events extracted from newswire reports. After summarizing related work on coordinated multiple views and multidimensional visual data analysis, we describe the cross-filtering technique and a general method for designing cross-filtered multiple view visualizations of tabular data sets. Using additional examples of cross-filtered visualizations of data from three other domains, we proceed to describe how cross-filtering can be flexibly customized with respect to data types across multiple dimensions, incorporated into more wide-ranging multiple view designs, and modularized for reuse across designs. We conclude by considering utility, usability, scalability, and future work.

2 EXAMPLE: WORLD EVENTS

The Kansas Event Data System (KEDS) uses automated extraction and encoding of English-language news reports to generate political event data [19]. The following research question motivates political scientists to generate and analyze such data sets: What temporal and geographic patterns of political activity can be discovered in international events reported by major news services? This question

*e-mail: cweaver@psu.edu

IEEE Symposium on Visual Analytics Science and Technology
October 21 - 23, Columbus, Ohio, USA
978-1-4244-2935-6/08/$25.00 ©2008 IEEE

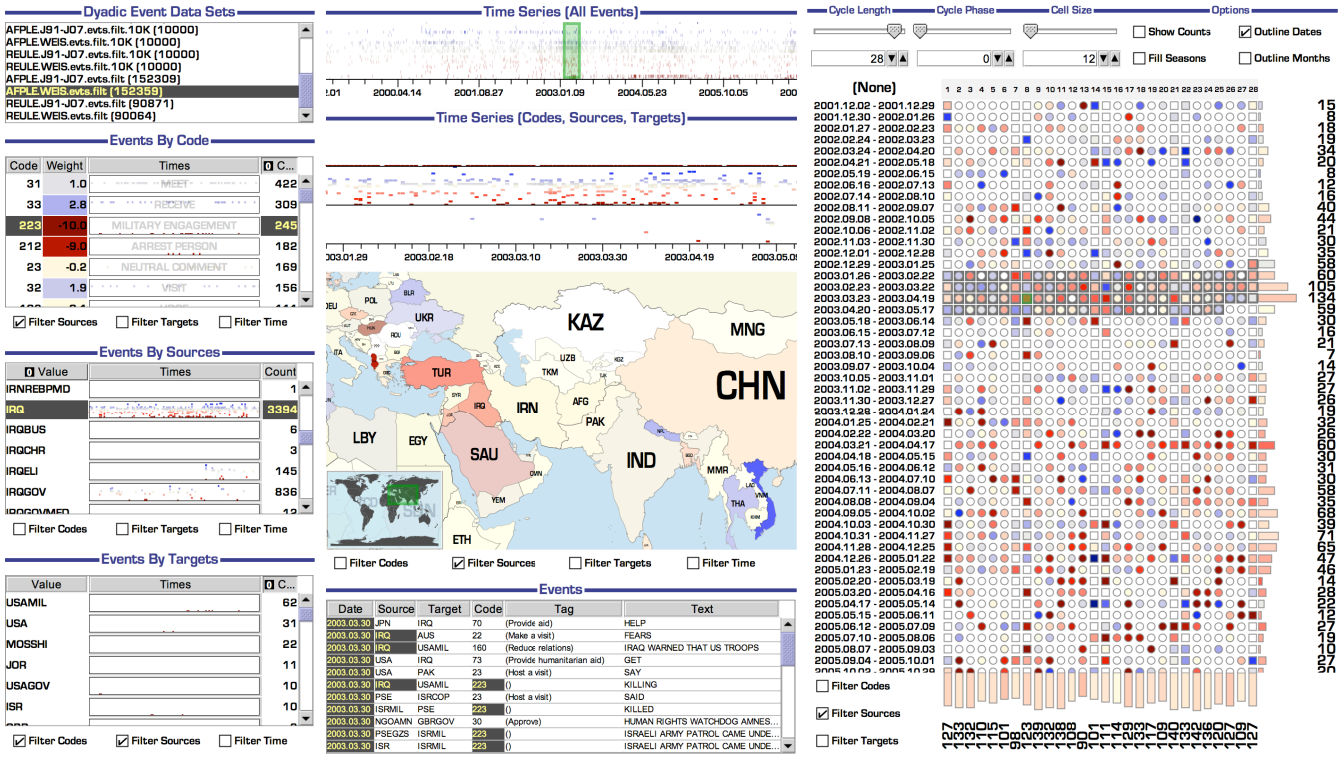

Figure 1: Cross-filtered visualization of geographic and temporal patterns in 150,000+ citations of political activity in international events reported by Agence France-Presse from May 1991 to January 2007. Cross-filtering on event source actor Iraq reveals a spike in conflictual events in early 2003. Further cross-filtering with military engagement as the chosen event type reveals the United States military as a frequent target actor.

is also of substantial practical interest to intelligence analysts, policy advisors, journalists, and social scientists who focus on political relationships between international actors. Such actors may include the governmental, military, diplomatic, health, and educational institutions of individual states as well as international organizations, resistance movements, and terrorist groups.

State-of-the-art statistical methods of political event analysis are generally quite useful for detecting coarse-grained spatiotemporal patterns, and can be targeted through manual effort at patterns involving particular kinds of events and sets of actors. What statistical methods lack is the immediacy of visual tools that enables analysts to flexibly drill-down in order to seek out the proverbial "needle in a stack of needles", that critical but subtle pattern of political activity hidden in a mountain of data.

Conversely, visual tools for political event analysis must be able to handle the large, high-dimensional data sets that statistical methods process with aplomb. A typical data set, of Agence France-Presse daily reports from May 1991 to January 2007, contains 150,000+ records representing 100+ kinds of events between 650+ source actors and 600+ target actors. Moreover, the data queries, visual representations, and user interactions in visual tools must be usable and useful even in the face of vastly more data than could possibly be displayed on the screen at once in a coherent manner.

We used the Improvise visualization environment [27] to design and implement an interactive visual tool (figure 1) for exploring and analyzing KEDS data sets. The tool displays three tables that summarize events over time for each event code (type of event), source actor, and target actor. A red-to-blue color gradient indicates the conflictual or cooperative political weight of events, as determined by their codes; "neutral" events appear in beige tones. (KEDS uses the World Event/Interaction Survey (WEIS) coding scheme to map each event onto a -10-to-10 weight scale that represents a spectrum from "conflictual" to "neutral" to "cooperative". [12]) A zoomable world map colors states (countries) according to the average weight of their events. A scrollable, wrapping calendar colors dates similarly, indicating overall number and political weight of events for individual dates, with colored histograms for rows and columns.

Figure 1 shows an example of one point in an analysis inquiry. The analyst has indicated an interest in Iraq by selecting "IRQ" as an event source. Filtering the map on sources reveals a strongly conflictual overall character of events involving target actors Turkey and Algeria, as well as a moderately conflictual overall character of events involving Saudi Arabia and Jordan. Filtering the calendar on sources reveals a rapid increase in reports of generally more conflictual events from February to April 2003. Filtering the code table view on sources reveals that over 16 years there have been 245 reports of military engagement in which Iraq was the source actor. Filtering the targets table view on sources and codes indicates that the United States military was reported as the target 62 times, with Shia Moslems and the governments of the United States, Jordan, and Israel reported as other frequent target actors.

Analysts can drill-down into events by selecting arbitrary subsets of codes, sources, targets, and dates, then cross-filtering on those subsets in different views. As a result, analysts can ask specific questions about relationships between groups of actors, kinds of actions, and patterns of events over time. Asking sequences of questions involves drilling "sideways" by selecting and deselecting items, turning filters on and off, and panning and zooming in the map and calendar. As it turns out, this style of interactive design can be generalized to many tabular data sets. In fact, the KEDS visualization was designed and implemented in Improvise in under a week using relatively minor variations of the queries and visual representations developed for visualization of historic hotel visitation patterns (figure 2) [26].

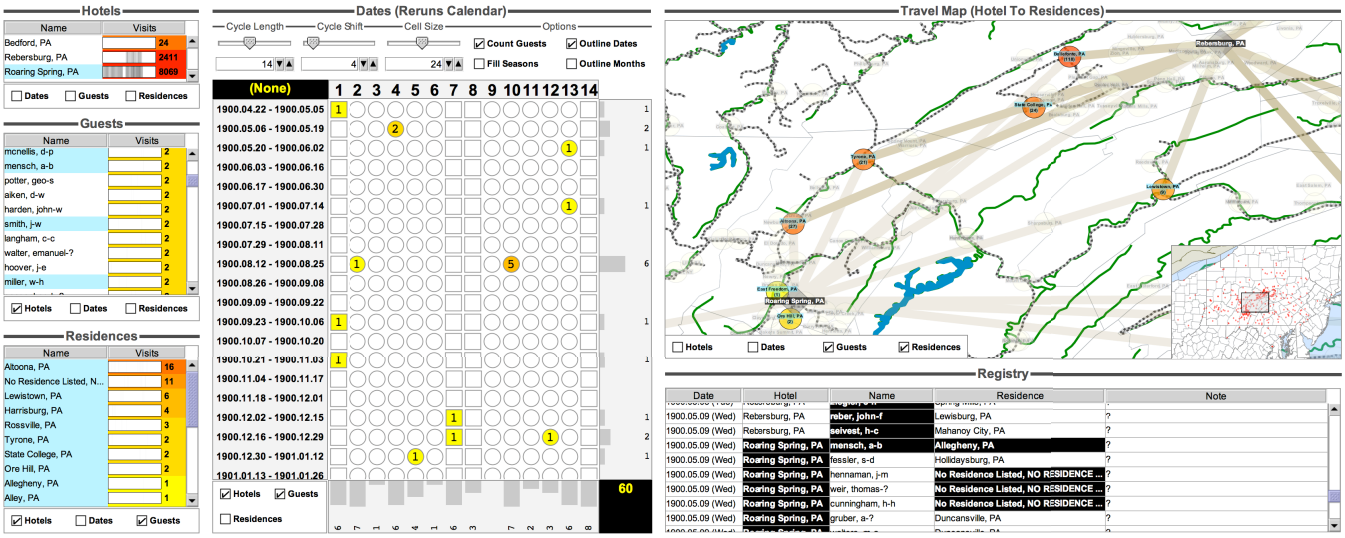

Figure 2: Cross-filtered visualization of guest registries from historic hotels in central Pennsylvania. Historical geographers explore early commercial travel patterns by posing sequences of who, what, where, and when questions involving arbitrary groups of hotels, guests, residences, and dates. Cross-filtering first on one hotel and then another allows identification of guests who visited both, with relevant dates and residences.

3 Related Work

Design patterns [7] are a well-established way to formalize the design of software artifacts, especially those that provide interactive user interfaces. The design of visualization tools is no different, often involving many of the same patterns, such as Model-View-Controller [10]. There are, however, multiple patterns that are widely used in and unique to visualization tools, making them significantly different from interactive software in general. Heer describes a collection of such visualization-specific patterns and their relationships [8]. Of particular relevance here are the Reference Model (e.g. the data state model [9]), Expression, and Dynamic Query Binding [3] patterns that are frequently used to implement various kinds of coordination in visualization tools.

Individual coordination techniques can themselves be described in terms of general patterns that combine navigation and selection [14]. The form and function of coordination patterns as individual building blocks is generally well understood in information visualization. Such coordination patterns serve as recipes for composing views, queries, and their interdependencies into coherent interactive visual representations [25]. The vast majority of tools being produced by the information visualization and visual analytics research communities employ the same combinations of these simple patterns over and over. As such, utilization of coordination patterns in the design of new tools remains mere craft, pursued in an effective but ad hoc manner by visualization experts.

The emerging challenge is thus to discover and formalize higher-order constructions from a well-known set of "atomic" visualization components, and to identify which constructions are manifestly both highly useful and usable for visual data exploration and analysis. Such constructions do exist in visualization research and practice, and can be taken as inspiration in this search.

XmdvTool [23] supports cross-dimensional analysis of data in multivariate visualizations using multidimensional brushing, such as an N-dimensional hypercube brush in a scatter plot matrix. Structure-based brushing [5] extends the idea from brushing in the orthogonal space in which data is displayed to the structured space (such as hierarchical) of the underlying data abstraction. A key difference between cross-filtering and these techniques is that of objects versus space; clicking, lassoing, rubberbanding, and other forms of brushing serve to select data items rather than a spatial region that contains them. This subtle distinction is a critical visualization design consideration whenever the data or visual abstraction depends on coordinated interaction in ways that can change the presence or position of data items in visual space.

IVEE/Spotfire [2] automatically matches sliders to data attribute types, thereby creating visualizations in which a central view can be interactively filtered by selecting subranges of data attribute values. One way to think of cross-filtering is as an extension of dynamic queries in which the sliders are all views that can independently filter each other at the analyst's discretion. Interestingly, cross-filtered visualizations can usefully preserve the central view as a terminal detail view that either hides or highlights data items as a function of matching selected attribute values in cross-filtered views.

Multiscale visualization using data cubes [21] formalizes a hierarchical multidimensional drill-down approach. The Polaris/Tableau/ShowMe [11] software provides an easy-to-use exploratory visualization builder interface based on drag-and-drop editing of data attribute hierarchies. Cross-filtering differs from this style of multiscale visualization in three ways. First, it is based on a single level of aggregation, namely grouping of unique attribute values. Second, it can freely incorporate derived attributes calculated from other attributes (even across tables). Third, it can parameterize its data and visual abstraction operations—including grouping, filtering, and visual encoding, on base or derived attributes—in terms of interaction in any view or slider, allowing significant variations in visualization design as needed to accommodate particular data structures or analysis requirements. Our goal for cross-filtering is complementary to multiscale visualization in that we aim to facilitate exploration of relationships that are complex, underrepresented in the data, and largely dependent on the experience and imagination of the analyst, and hence are hard to expose through hierarchical aggregation of context-free data types alone.

Jigsaw [20] and the contemporaneous hotels visualization are domain-specific tools that use similar variations of cross-filtering for interactive analysis. The visual abstractions that Jigsaw uses for individual data attributes are relatively simple but cross-linked table views. Useful features in Jigsaw include dynamic picking of data attributes for cross-filtering, the ability to sort selected items to the top of cross-filtered table views (which we have since added to our own table views), and a simple graph view for exploring

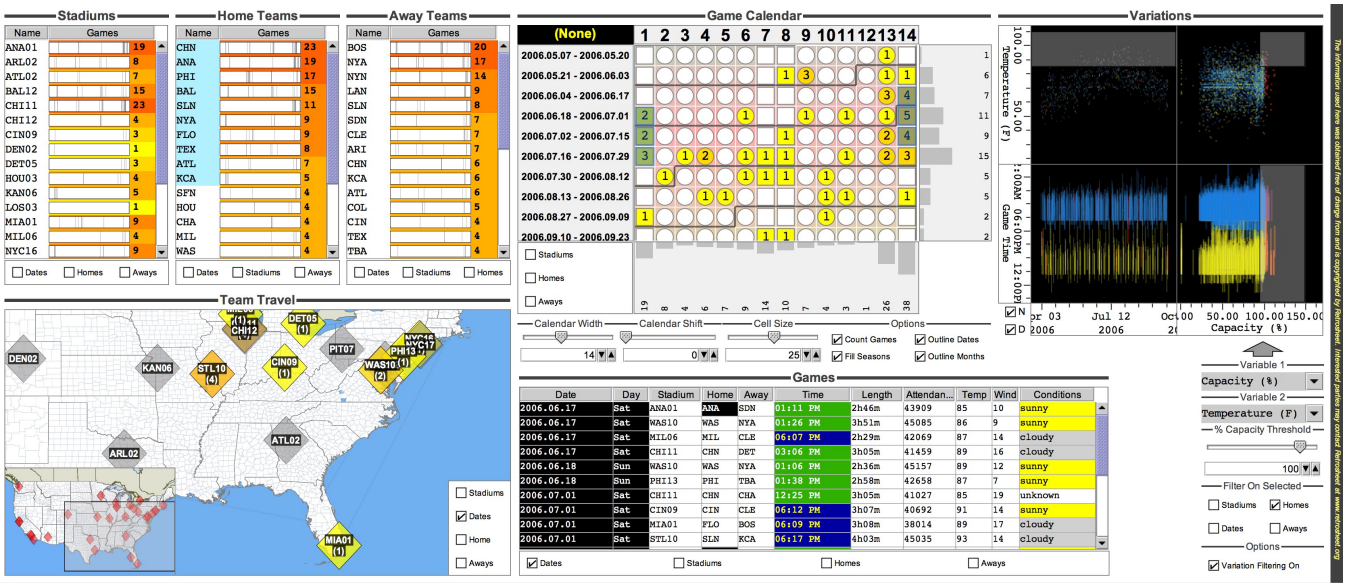

Figure 3: Visualization of 2001-2006 major league baseball games recorded in the Retrosheet database [1], with cross-filtering on stadiums, home teams, away teams, and game dates. Variation filtering provides additional globally switchable cross-filtering on ranges of game time and two user-picked numeric attributes in a scatter plot matrix, here revealing high attendance 2006 weekend games with temperatures over 95°F.

connections between attribute values. With the quite useful exception of attribute picking, variations on cross-filtering in support of particular data sets and analytic tasks happen in Jigsaw through development by visualization experts using regular programming, in contrast with rapid and flexible exploratory design of data and visual abstractions in Improvise.

Visualization schemas [13] build upon the Snap model to support different kinds of coordinated interaction between pairs of views—including coupled loading of data, selection of items, and navigation over items—in terms of one-to-one, one-to-many, and many-to-many relationships between relational data sets. Any two views can be coordinated using a compound join to associate different attributes in a many-to-many fashion. Cross-filtered views in its simplest form can be thought of as a visualization schema in which the data abstraction consists of compound joins that perform independent, switchable, many-to-many filtering of selected items between pairs of views. Particular data sets and analytic needs, however, generally call for sometimes significant variations in coordination, data abstraction, and visual abstraction that are hard (if not impossible) to capture in high-level models like visualization schemas.

4 Cross-Filtered Views

In this section, we consider cross-filtered views from three different perspectives: (1) as a method for interactively expressing sequences of multidimensional set queries by selecting and filtering unique data values across pairs of views; (2) as a general pattern for constructing an interdependent set of data transformation operations that support the method; and (3) as an open-ended space of design variations for instantiating the pattern in particular visual analysis applications. We describe each perspective in the context of example visualizations designed to support analysis in different information domains.

4.1 Interaction

When an analyst is tasked with finding a needle in a stack of needles, a good strategy is to start by examining needles with characteristics similar to those of the needle being sought. Because it is the analyst's knowledge and experience that drives the choice of characteristics worthy of examination, this strategy can work even when the needle is unknown, and the task is discovery. The corresponding goal of cross-filtered views is to facilitate the identification and characterization of relationships between people, places, times, and other values in multidimensional information through visual interaction. As such, it is first and foremost a method for using visualization to ask detailed questions about correspondences between data item characteristics.

The structure of a cross-filtered visualization is made up of three essential elements: views, brushes, and switches. Each view displays the unique values of a particular data attribute in a type-appropriate way, e.g. a table view for names or a calendar for dates. Each brush selects a subset of the values displayed in a single view. Each switch toggles filtering between a directed pair of views. The semantics of filtering is that of simple association: *show only those values in view B that co-occur in the data with the values selected in view A*. The example visualizations shown in figures 1–4 use variations of this structure to support analysis in different domains.

The mechanical process of cross-filtering consists of nothing more than sequences of interleaved brushing and switching interactions provoked by observation of visible values in views. It is the crucial inclusion of the analyst's observations, however, that makes the mechanics of cross-filtering deceptively simple. Interactions are expressions of questions (during exploration) or hypotheses (during analysis) about the unobserved characteristics of a set of entities as a function of their observed characteristics, and what any associations in those unobserved characteristics imply about relationships between entities. The critical factor here is the knowledge, experience, and general perceptual and cognitive capabilities that the analyst brings to bear on the interpretation of current interactive states. Consequently, the mechanical process of cross-filtering is simply a proposition externalization framework for an inductive reasoning process that includes: (1) informed examination of the characteristics of people, places, times, etc. to identify potentially related subsets; (2) selective collection of values into subsets to manifest them as a coherent group cognitively and computationally; (3) abridgement of the scope of visual contexts in which further questions or hypotheses involving other characteristics may be considered.

Working with cross-filtered views is like sifting particles through a sequence of screens in which the holes have shapes that match

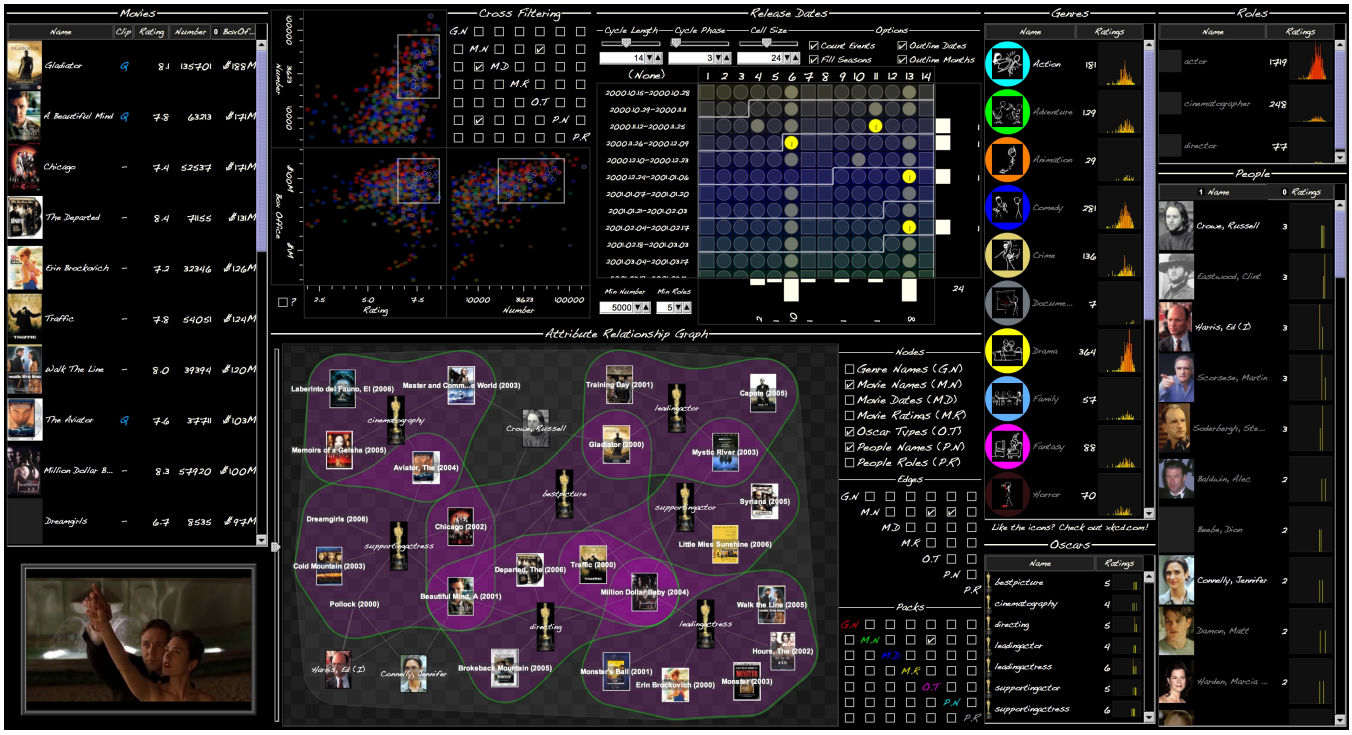

Figure 4: The Cinegraph visualization [24] of recent popular movies in the Internet Movie Database, with cross-filtering of seven dimensions (movies, ratings, release dates, genres, awards, people, and roles) spread across four data sets. Cross-filtering from awards to movies to people reveals winning collaborations between top actors and directors. Selected attribute values populate a graph that shows individual relationships.

the particles and their characteristics that are of particular interest. Expression and consideration of complicated questions/hypotheses happens through repetition and interleaving of steps. Because selections can be modified and cross-filters can be toggled for any attribute at any point in the process, questions/hypotheses can be restated quickly and flexibly by adding, removing, changing, and reordering dimensional clauses, all performed by brushing or switching. As a result, trained analysts can follow complex lines of inquiry using sequences of simple interactions to visually interrogate data about who, what, where, and when by quickly drilling down into arbitrary subsets of multidimensional information. Cross-filtering is not only one instance of a potentially large class of high-level coordination patterns, but also an instance of a similarly large class of strategies for supporting cognition through visual interaction.

4.2 Queries

Cross-filtered visualizations build upon a graph of data transformations that connect views, brushes, and switches. Each view performs four transformations on the input data: (1) grouping (γ) of data records into sets for each unique attribute value; (2) filtering (ϕ) of each set, keeping records whose attribute values match those brushed in other views; (3) projection/visual encoding (π) of each value and its filtered set; (4) selection (σ) of values/sets corresponding to brushed glyphs in the view. Each projection uses the selection in its own view to highlight brushed glyphs. Each filter either ignores or applies selections of other views to cross-filter its own view's sets, depending on the state of the corresponding switches.

For example, figure 5 shows the graph of data transformations used in the hotels visualization. The concatenated entries from several hotel guest registers (T) are grouped by hotels (h), guests (g), residences (r), and dates (d). Each group (G) is filtered (G') then projected/visually encoded (V) in order to populate the hotels table view, guests table view, residences table view, and cyclic calendar view. The analyst drills down by brushing subsets of values in the four views and by switching cross-filtering between views (using checkboxes at the bottom of each view to toggle incoming filtering from each of the other views). This symmetric and relatively simple organization of relationships between data abstraction, visual abstraction, and coordination allows the analyst to express cross-filtering queries in a uniform manner across multiple data attributes, regardless of how each view visually encodes and brushes the unique values of its particular attribute type.

4.3 Design Variations

The query strategy used in cross-filtering has evolved over time through experimentation with the design and operation of several visualizations. Although the current basic organization of data transformations is concrete and may be reused unchanged in new visualizations, it is far from rigid. Grouping, filtering, and projec-

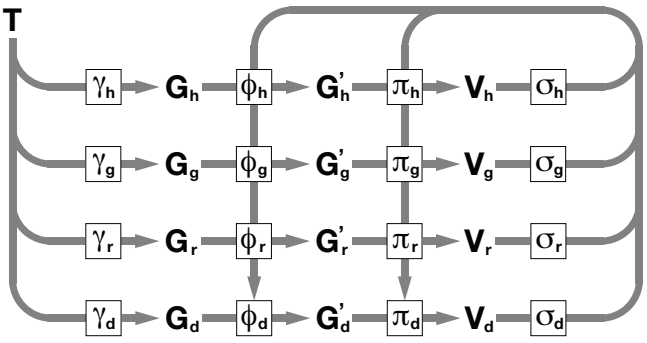

Figure 5: Cross-filtering transformations in the hotels visualization.

		KEDS	Hotels	Retrosheet	Cinegraph
Attributes	**Nominal**	code (event)	name (guest)	name (home team, away team)	name (movie, genre, oscar, person, role)
	Temporal	date (event)	date (visit)	date & time (game)	date (release)
	Spatial	region (countries)	location (hotel, residence)	location (stadium)	-
	Numerical	cooperative/conflictual weight	-	capacity, attendance, temperature, wind speed	box office, rating average, rating count
Auxiliary Views	**Pre-filter**	list (data sources)	-	-	sliders (ratings & roles thresholds)
	Post-filter	map (world)	map (Pennsylvania)	map (North America), rich drill-down table	attribute relationship graph
	Detail	drill-down table, split time series	drill-down table	-	movie viewer
	Nested	scatter plot (date vs. weight)	1-D heatmap (visit count by date)	1-D heatmap (game count by date)	histogram (rating distribution)

Figure 6: Variation in data attributes and auxiliary views in the four example cross-filtering visualizations.

tion operations can be customized individually or in combination as called for while designing visualizations for particular data sets and analysis tasks. In particular, visual encodings are essentially unconstrained and may be specialized to suit the character and distribution of each data dimension (figure 6), and as such may have a practically unlimited number of distinct, useful variations. To flesh out the cross-filtering pattern, we briefly describe more extensive structural variations in the design of the example visualizations.

4.3.1 Visual Abstraction

The only requirement that basic cross-filtering imposes on visual abstraction is that unique attribute values have individually brushable representations. Views can map values into graphical attributes using nearly any visual encoding technique, so long as it is possible to interactively select any arbitrary set of values, and that some form of differential visual encoding distinguishes selected items from unselected ones. Moreover, it is often useful to encode the grouping set associated with each value. A particularly useful way of showing information about groupings for each dimension is to visually encode rows in table views using small multiples of nested time series, scatter plots, heatmaps, or histograms. Entire views can be replicated (using homologous filter and projection queries) to enable parallel analyses involving multiple independently switchable brushes on each attribute. Similarly, multiform visualization [18] allows users to adopt their own analysis strategies by cross-filtering the same attributes displayed in multiple views in different ways, as with the residences table view and map in the hotels visualization. Selections in multiform views may be coupled or independent.

4.3.2 Data Abstraction

All of the visualization designs that we have built around cross-filtering so far have involved not only substantial customization of visual encodings for each data dimension—often even when these dimensions are semantically the same, e.g. the different ways of encoding dates in various calendar views—but also changes to the organization of core query operations. More extensive structural changes involve extended designs that contain auxiliary views and sliders that: (1) prefilter the full data table(s) prior to cross-filtering; (2) provide alternate geographic or temporal representations of the data, filtered on selections in a subset of the cross-filtered views; or (3) provide a variation of details-on-demand in which records are highlighted (rather than filtered out) in a detail view as a function of selection of their attribute values in cross-filtered views.

For example, the data abstraction used in Cinegraph (figure 7) varies from the standard method in two major ways. First, the movies database consists of seven dimensions split across four relational data tables in a simple star schema with an integer movie identifier as primary key. We adapted cross-filtering to use both intra- and inter-table cross-column indexing, effectively treating the four tables as a single cross-product table for cross-filtering purposes (but without the added space or time complexity of a precomputed full join). While straightforward to do, we have not yet adapted cross-filtering for foreign-foreign key relationships needed by more complex schemas. For instance, cross-filtering people on roles in Cinegraph is counterintuitive because it calculates the set of people who were in movies that had someone playing any of those roles, rather than the set of people who played any of those roles in some movie. It is critical to consider carefully the semantics of all attributes pairwise when designing cross-filtered visualizations.

Second, we dealt with interactive performance limitations due to large data size in Cinegraph by adding two prefiltering sliders that perform an adjustable amount of online preprocessing controlled by two hidden interactive parameters: a number of ratings threshold that filters out references to infrequently rated movies from all four tables, and a number of roles threshold that further filters the people table. Because cross-filtering involves simple tables, data abstractions can be extended to include an arbitrary amount of data preprocessing, whether interactively driven or not.

4.3.3 Attributes

Because cross-filtering is based on unique attribute values, it can work with any attribute type by treating all values as nominal measurements. Nevertheless, temporal, spatial, and other numeric attribute values are often more useful for analysis if they are preserved as ordinal measurements. This is typically a simple matter of matching attributes with views designed for them in the usual way, e.g. calendars for dates and maps for geographic regions.

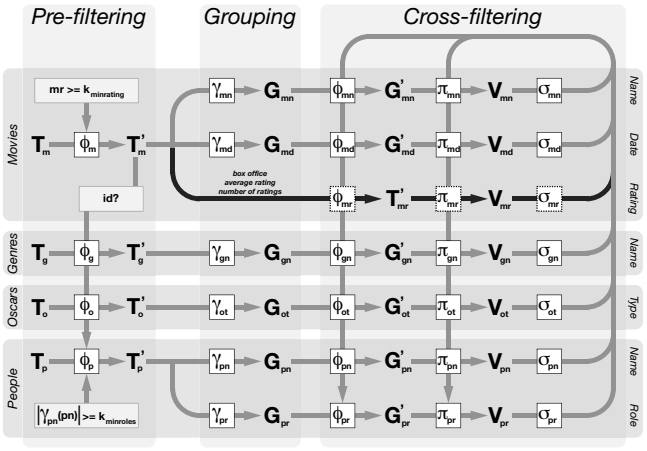

Figure 7: Cross-filtering queries in the Cinegraph visualization. Inputs into the grouping stage consist of four independent tables, prefiltered to include only high-rated movies and frequent actors. The ratings "dimension" is a 3-D space of movie statistics displayed in a scatter plot matrix with rubberband brushing.

The preponderance of multiple secondary numerical attributes in many data sets makes it generally impractical to support full cross-filtering across all dimensions simultaneously. In such cases, attributes can be collected into a single ungrouped compound attribute for purposes of cross-filtering. For instance, the Cinegraph visualization treats box office sales, average IMDB user rating, and number of IMDB user ratings as a single attribute displayed in a scatter plot matrix with rubberband brushes. Cross-filtering tests for 3-D rubberband containment rather than individual item selection. Conversely, all three ratings plots apply the same filter (using modified indexes to accommodate ungrouped attribute columns) to the data prior to application of slightly different 2-D projections. (Although the rubberbands look and act like a 3-D rectangular brush in XmdvTool, switchable filtering give users control over brushing-highlighting semantics during analysis, and thus over paths of dimensional drill-down to examine and follow.)

Flexible data abstraction means that cross-filtering can involve both raw and derived data attributes. Derived attributes generally involve one-to-one record transformations for purposes such as merging related data columns (such as last-first-middle name concatenation in the hotels visualization), clustering of numerical values in high-cardinality dimensions (such as rounding average ratings to one-tenth values in the Cinegraph visualization), and even hierarchical categorization of values into multiscale attributes.

4.3.4 Multiscale Visualization

Multiscale visualization allows analysts to see different amounts of detail by zooming in and out. For example, DataSplash [15] allows analysts to see different amounts of detail in a view by changing the zoom "altitude" in a layer manager. Zooming in Tableau is less literal, involving transitions between natural and artificial levels of aggregation in geospatial (state, county, etc.), temporal (year, month, day, etc.), and nominal attributes. Although we have not yet designed an example of multiscale cross-filtering, there are at least two ways to do it. The first way would be to use cross-filtering that depends on altitude in a view with semantic zooming. Changing zoom levels would not only change the appearance of the navigated view, but also filter other views to show only values for items selected at the new zoom level. Selections might even be translated across levels by applying union or intersection semantics across different levels of aggregation. The second way would be to calculate multiple derived attributes for each scale of interest, treating them as if they were independent for purposes of cross-filtering. For instance, all four example visualizations would be more useful if it were possible to filter all views (including calendars) on a weekday attribute derived from dates. This approach would allow analysts to pose complex cross-scale questions without the screen space limitations or imposed type constraints of most hierarchical techniques. For instance, baseball enthusiasts could ask questions in the Retrosheet visualization about weekend games in the first month of the season in the 60's and 70's, using Day-of-Week, Month, and Decade views to select sets of derived attribute values, sets that constitute ad hoc temporal categories in the user's imagination.

5 Discussion

The design variability and broad analytic scope of the cross-filtering technique makes it a challenging evaluation target, above and beyond the difficulties inherent in evaluating visualization tools in general [17]. In particular, evaluation of cross-filtering in the context of specific tools is feasible only if we are able to recruit both experts who can longitudinally validate usability and usefulness for analysis in the relevant knowledge domains (international politics, historical geography, baseball, movies) as well as formative study participants who have at least passing knowledge of those domains.

Development of the hotels visualization benefitted substantially from close collaboration with the domain experts who collected the registry data for the purpose of exploring hotel guest visitation behavior, producing extensive longitudinal feedback on the visualization design [26] and contributing to a doctoral dissertation in historical geography [6]. Indeed, it was discussion of particular analytic needs in this domain that led to discovery of the cross-filtering technique in the first place. An early design using the technique underwent an evaluation in which a group of geography students used the HERO e-Delphi web portal system [16] to provide qualitative feedback about the hotels visualization, in response to both prescribed analysis tasks and free form exploration involving cross-filtering.

We experienced two major surprises during evaluation of the cross-filtering method. First, utility benefits from the ability of analysts to mitigate noise, damage, redundancy, deception, ambiguity, and related forms of uncertainty in data by intelligently aggregating similar-seeming attribute values. Indeed, uncertainty is itself often useful in context. Uncorrected transcription of faded, handwritten guest names in the visualized data helped to preserve some of the analog character of the historic hotel registries, allowing expression and analysis of more subtle hypotheses about visitation patterns.

Second, usability suffers from a frequent "out of sight, out of mind" effect that occurs when selected items both cross-filter other views and are themselves cross-filtered out. Cross-filtering relies on idempotency of selections, lest a single switch set off a multiview cascade of irreversible selection-filtration set intersections until a fixed point or null set is reached, destroying the analyst's ad hoc attribute groupings along the way. Expert and student feedback, while generally positive, revealed concerns about the ability to see and remember more than the current state in the analysis process, prompting suggestions for functionality to help guide cross-filtering interaction by capturing and visualizing steps in the analysis process itself. We are also exploring techniques that employ compound brushing [4] to preserve visual context by replacing or supplementing filtering with multivariate "cross-highlighting" of items.

Achieving reasonable scalability is a key objective of the query strategy used in cross-filtering. For cross-filtered views that have simple visual encodings, the Improvise in-memory query and rendering engine typically can support direct manipulation ($<$100ms) interactivity with upwards of 100K data attribute values (rows times columns) on typical current desktop hardware (2GHz dual core processor, 2GB memory, 128MB video). Extremes in the dimensionality, cardinality, and visual complexity of data processing that arise during cross-filtering make it difficult to estimate even loose upper bounds on interactive performance.

Cross-filtering supports many design variations. However, visualization builders are not relieved of the responsibility for good design. Particular information domains, analytic tasks, and data sources call for prudent choices in organization and preprocessing of input data sets, selection of raw and derived attributes to cross-filter, visual encoding of attribute values, and coordination with auxiliary views. In designing all four example visualizations, we found it necessary to perform at least some offline data preprocessing in order to clean up and transform the original data sources into suitably canonical tabular form. It was often possible to reduce the dimensionality of the data once the cross-filtered attributes for the visual analysis tool had been chosen carefully. In the Cinegraph visualization, it was also necessary to minimize the number of movies and people to achieve a minimum level of interactivity. Except for the original hotels visualization, the time to develop each of the example visualizations in Improvise was on the order of a week. This time was split roughly equally between data preprocessing, actual live design of the visualization interface, and interactive exploration and analysis of the data. Much of this speed resulted from saving the cross-filtering queries in the hotels visualization as a reusable, schema-independent template that can be loaded into new or existing visualizations to add cross-filtering functionality. A nice feature of this template is a switch/checkbox permutation matrix (figure 4,

just left of top center) in which each row determines filtering of an attribute and each column determines filtering by an attribute.

Visualization development in Improvise follows a user-centered model in which domain experts work closely with visualization designers to create and coordinate multiple views rapidly and iteratively as needed during visual data analysis. Unlike other toolkits and systems, Improvise enables designers to discover new combinations of visualization components and modify existing ones through interactive exploratory design. The goal of populating a space of design strategies for sophisticated visual analysis tools is nevertheless system-agnostic. Despite implementation in Improvise, the cross-filtering model and design pattern are generalizable, and can be adapted to other visualization toolkits and systems.

Future work will look at ways to extend Improvise for semiautomatic construction of cross-filtered visualizations. Users would choose data attributes of interest, then associate each attribute with an appropriate view type. Although initial design would occur in a high-level interface, advanced users could choose to continue working directly in the Improvise query builder to customize their visualizations as needed for deeper or more specialized analysis.

6 Conclusion

Cross-filtering is a method and design pattern for fast and flexible interactive visual drill-down into fine-grain relationships buried in information spread across multiple data sets. Multiple example visualizations demonstrate the generality and flexibility of cross-filtering, and provide insights into specialization for different data sources and analysis needs. As a result, we have made substantial progress in our ability to discover, instantiate, and reuse effective multidimensional visual analysis designs at a high level of design abstraction. By discovering and understanding designs like cross-filtering, we hope to provide analysts with the means to seek out and dissect subtle patterns in multidimensional information spaces. Moreover, we hope to develop design principles that are generalizable to different information sources, extendable to multiple interrelated data sets, and rapidly combinable and customizable into functionally-rich visual tools for far-reaching visual data analysis.

Acknowledgements

Thanks to David Fyfe, Deryck Holdsworth, Alan MacEachren, Donna Peuquet, Anthony Robinson, Phil Schrodt and many others who contributed to development of the visualizations in this paper.

This work was performed with partial support from the National Visualization and Analytics Center (NVAC), a U.S. Department of Homeland Security Program, under the auspices of the Northeast Regional Visualization and Analytics Center (NEVAC). NVAC is operated by the Pacific Northwest National Laboratory (PNNL), a U.S. Department of Energy Office of Science laboratory.

References

[1] Retrosheet. http://www.retrosheet.org/, March 2008.

[2] C. Ahlberg. Spotfire: An information exploration environment. *SIGMOD Record*, 25(4):25–29, December 1996.

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

[4] H. Chen. Compound brushing. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 181–188, Seattle, WA, October 2003. IEEE Computer Society.

[5] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):150–159, April-June 2000.

[6] D. Fyfe. *Commerce and Sociability in Small-Town America: Explorations in Historical GIScience*. PhD thesis, The Pennsylvania State University, University Park, PA, January 2008.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. 1st edition, October 1994.

[8] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):853–860, September/October 2006.

[9] E. H. hsin Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 63–70, Research Triangle Park, NC, October 1998. IEEE Computer Society.

[10] G. E. Krasner and S. T. Pope. A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49, August 1988.

[11] J. D. Mackinlay, P. Hanrahan, and C. Stolte. Show Me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, November/December 2007.

[12] C. McClelland. World event/interaction survey (WEIS) project, 1966-1978 [computer file]. University of Southern California. 3rd ICPSR ed. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [producer and distributor], 1999.

[13] C. North, N. Conklin, K. Indukuri, and V. Saini. Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases. *Journal of Information Visualization*, 1(3-4):211–228, December 2002.

[14] C. North and B. Shneiderman. A taxonomy of multiple window coordinations. Technical Report CS-TR-3854, University of Maryland Department of Computer Science, 1997.

[15] C. Olston, A. Woodruff, A. Aiken, M. Chu, V. Ercegovac, M. Lin, M. Spalding, and M. Stonebraker. DataSplash. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 550–552, Seattle, WA, June 1998. ACM.

[16] W. Pike, B. Yarnal, A. M. MacEachren, M. Gahegan, and C. Yu. Infrastructure for human-environment collaboration: Building a prototype for the future of science. *Environment*, 47(2):8–21, 2005.

[17] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of ACM Conference on Advanced Visual Interfaces (AVI)*, pages 109–116, Gallipoli, Italy, May 2004. ACM.

[18] J. C. Roberts. Multiple-view and multiform visualization. In R. Erbacher, A. Pang, C. Wittenbrink, and J. Roberts, editors, *Proceedings of SPIE (Visual Data Exploration and Analysis VII)*, volume 3960, pages 176–185. SPIE, January 2000.

[19] P. A. Schrodt. *Event Data in Foreign Policy Analysis*, pages 145–166. Prentice-Hall, New York, 1994.

[20] J. Stasko, C. Görg, Z. Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 131–138, Sacramento, CA, October 2007. IEEE.

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

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

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

[24] C. Weaver. Infovis 2007 contest entry: Cinegraph. In *Proceedings of the IEEE Symposium on Information Visualization (Compendium)*, Sacramento, CA, October 2007. IEEE.

[25] C. Weaver. Patterns of coordination in Improvise visualizations. In *Proceedings of Visualization and Data Analysis*, pages 1–12, San Jose, CA, January 2007. SPIE.

[26] C. Weaver, D. Fyfe, A. Robinson, D. W. Holdsworth, D. J. Peuquet, and A. M. MacEachren. Visual analysis of historic hotel visitation patterns. In *Proceedings of the Symposium on Visual Analytics Science and Technology (VAST)*, pages 35–42, Baltimore, MD, October 2006.

[27] C. E. Weaver. *Improvise: A User Interface for Interactive Construction of Highly-Coordinated Visualizations*. PhD thesis, University of Wisconsin–Madison, Madison, WI, June 2006.