Cross-Filtered Views for Multidimensional Visual Analysis

Chris Weaver, *Member*, *IEEE Computer Society*

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. We also identify several important limitations of the approach. The demonstrated analytic utility of these examples suggests that cross-filtering is a suitable design pattern for instantiation in a wide variety of visual analysis tools.

Index Terms—Information visualization, interactive data exploration and discovery, coordinated views, multidimensional visual analysis.

1 Introduction

Engendering support for open-ended analytic reasoning through the development of new visual representations and interaction techniques is a key effort in the research agenda for visual analytics [1]. 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; and 3) users can rapidly toggle brushing filters between pairs of views —*show only those values in view B that co-occur in the data with the values selected in view A*—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; and 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

- *The author is with the School of Computer Science and the Center for Spatial Analysis, University of Oklahoma, 3100 Monitor Ave, Suite 180, Norman, OK 73072-7807. E-mail: weaver@cs.ou.edu.*

Manuscript received 16 Feb. 2009; revised 8 May 2009; accepted 16 July 2009; published online 7 Aug. 2009.
Recommended for acceptance by T. Ertl.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-2009-02-0036.
Digital Object Identifier no. 10.1109/TVCG.2009.94.

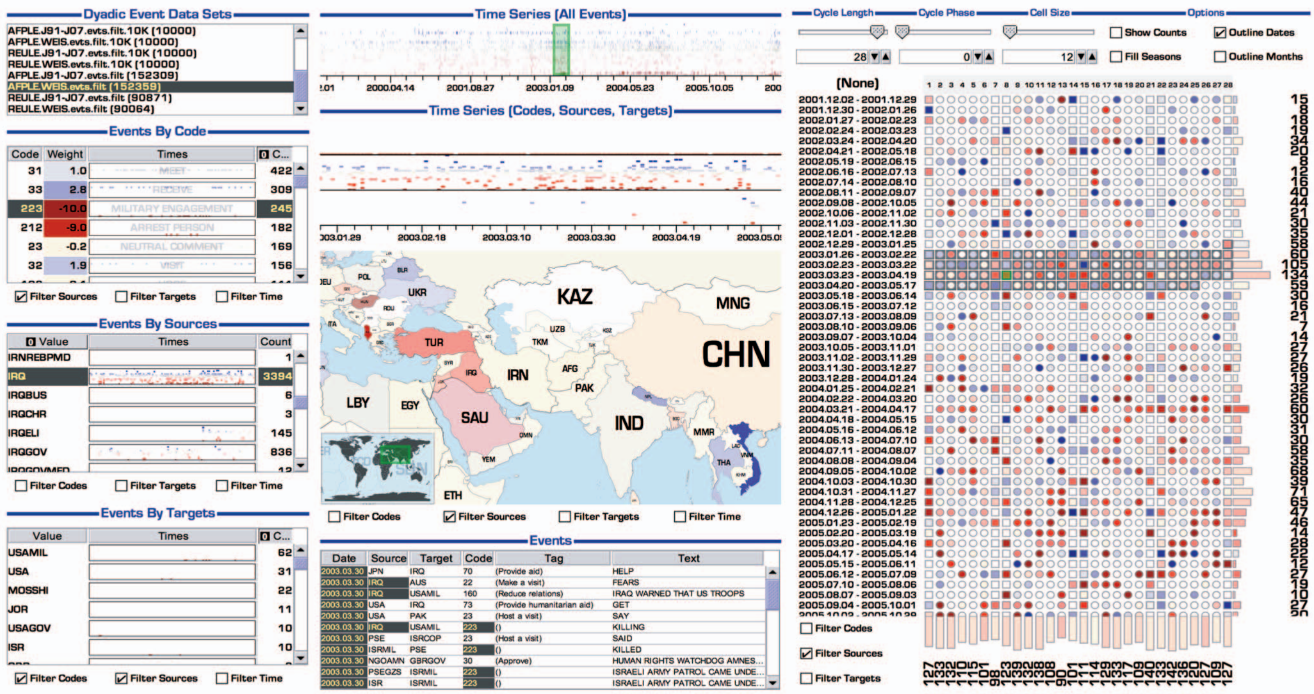

Fig. 1. Cross-filtered visualization of geographic and temporal patterns in 150,000+ citations of political activity in international events reported by Agence France-Presse from May 1991 to January 2007. Cross-filtering on event source actor Iraq reveals a spike in conflictual events in early 2003.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

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 multi-dimensional 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 four 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 key limitations of cross-filtering in terms of utility, usability, and scalability, and some of the future directions toward addressing these limitations.

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 [2]. The following research question motivates political scientists to compile and analyze such data: What temporal and geographic patterns of political activity can be discovered in international events reported by major news services? This question 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 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 [3] to design and implement an interactive visual tool (Fig. 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;

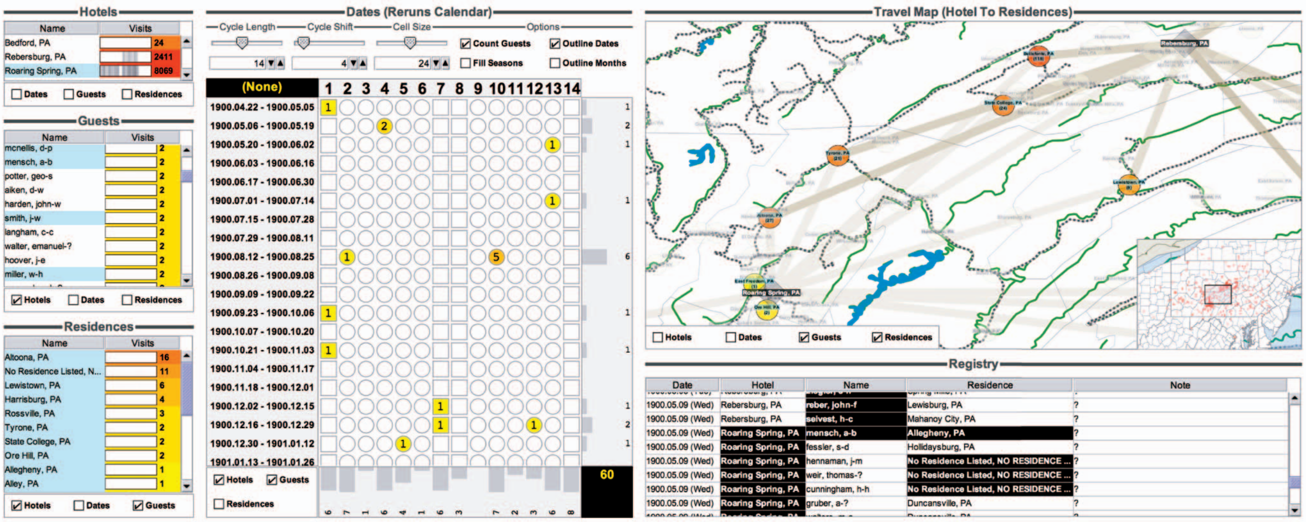

Fig. 2. Cross-filtered visualization of guest registries from historic hotels in central Pennsylvania. Historical geographers explore early commercial travel patterns by posing sequences of who, what, where, and when questions involving arbitrary groups of hotels, guests, residences, and dates. Cross-filtering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

"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." [4]) A zoomable world map colors states (countries) according to the average weight of their events. A scrollable, wrapping calendar colors dates similarly, indicating the overall number and political weight of events for individual dates, with colored histograms for rows and columns.

Fig. 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 target table view on sources and codes indicates that the US military was reported as the target 62 times, with Shia Moslems and the governments of the US, 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 within a week using relatively minor variations of the queries and visual representations developed for visualization of historic hotel visitation patterns (Fig. 2) [5]. (Improvise and the example visualizations in this paper are available online at http://www.cs.ou.edu/~weaver/improvise/.)

3 Related Work

Design patterns [6] 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 [7]. 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 and Agrawala describe 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 [10] 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 [11]. 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 [12]. 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 [13] 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 [14] extends the idea from brushing in the orthogonal space in which data are 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 design consideration whenever the data or visual abstraction depend on coordinated interaction in ways that can change the presence or position of items in visual space.

IVEE/Spotfire [16] 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 [17] formalizes a hierarchical multidimensional drill-down approach. The Polaris/Tableau/Show Me [18] 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 [19] 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 connections between attribute values. With the 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.

QlikView [20] is a commercial tool for visualizing associations across data columns in multidimensional relational databases. As with Jigsaw and cross-filtering, tools created using QlikView generally display most data attributes in a compact layout of table views, but can also display attributes in a variety of common visual abstractions such as maps and bar charts. In a QlikView table view, brushing highlights items using a row fill color and moves them to the top (subsorting within selected and unselected rows according to the natural ordering of the attribute data type). Cross-filtering differs from QlikView in four ways. First, cross-filtering does not specify any particular visual encoding of "highlighting" for brushed attribute values, in table views or otherwise. However, visual encoding in QlikView is likely a by-product of the implementation rather than any fundamental limitation in its overall approach. Second, whereas cross-filtering elides co-occurring values, QlikView uses an alternate row fill color to highlight them. Consequently, the two approaches can be generally distinguished in terms of zoom+filter versus focus+context. Third, QlikView imposes brushing constraints in which item selections in other views are automatically cleared if they do not co-occur with newly brushed items, in contrast with manual toggling of brushing constraints across attributes in cross-filtered views. Loosened constraints on brushing allow cross-filtering users to opt for greater interactive flexibility during exploration and analysis, at the expense of possible visual inconsistency in apparent key relationships between attributes. Overall, a careful comparative evaluation of visual representation and interaction in cross-filtered views, Jigsaw, and QlikView seems warranted.

Visualization schemas [21] 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 supports the method; and 3) as an open-ended space of

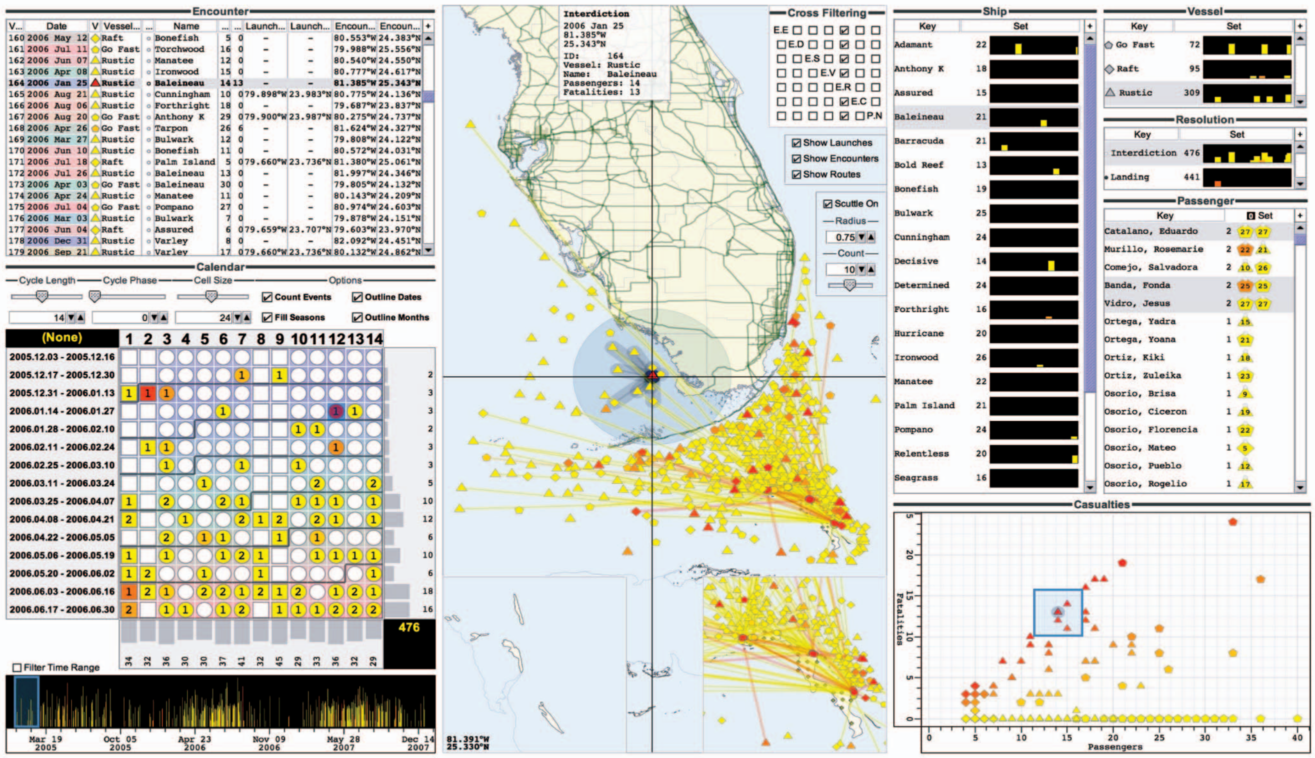

Fig. 3. Geovisualization of data from the VAST 2008 Migrant Boat Mini Challenge, with cross-filtering on encounter, date, interdicting ship type, migrant vessel type, resolution, casualties, and passenger names. The casualty "dimension" is a 2D space of encounter statistics displayed in a scatter plot with rubberband brushing. Two independent variation filters provide additional globally switchable filtering on 1) a range of dates in a time series plot (bottom left) and 2) a k-nearest-neighbor-within-great-circle-radius "scuttler" mouseover tool in a map (center).

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 Figs. 1, 2, 3, 4, and 5 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 characteristics of people, places, times, and so on to identify potentially related subsets; 2) selective collection of values into subsets to manifest them as a coherent group cognitively and computationally; and 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 the particles and their characteristics that are of particular interest. Expression and consideration of

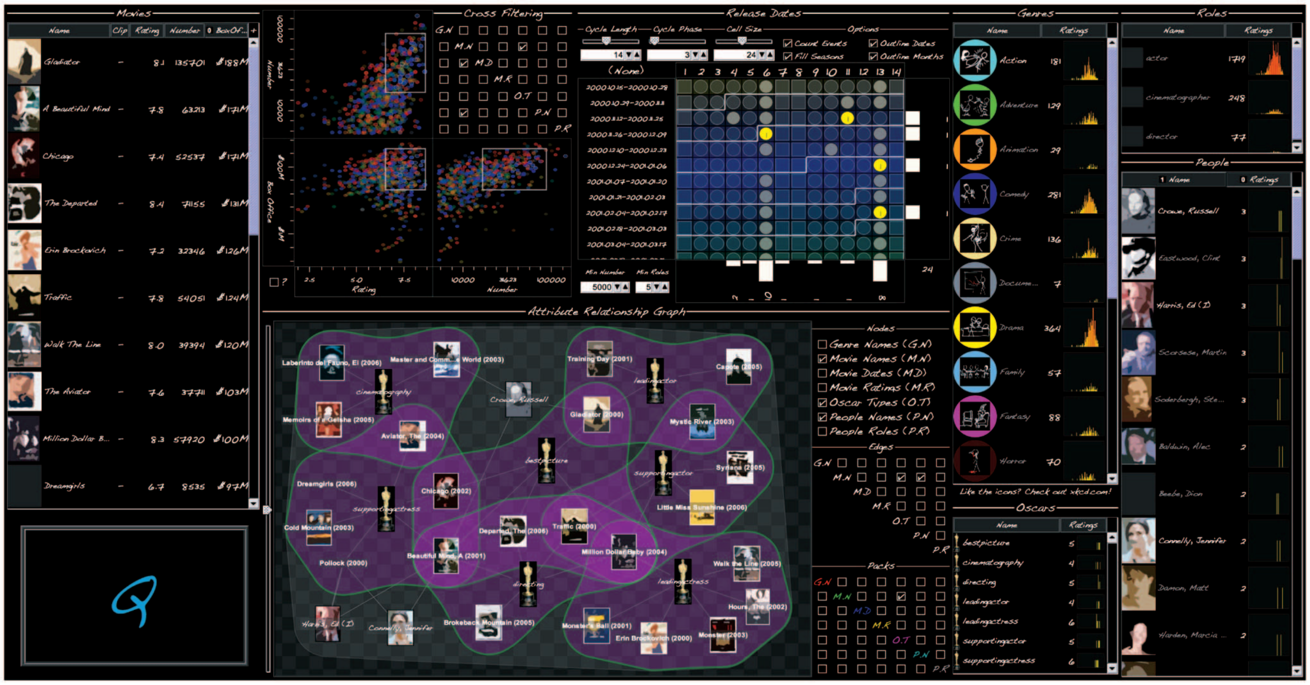

Fig. 4. The Cinegraph visualization [15] of recent popular movies in the Internet Movie Database (IMDB), with cross-filtering of seven dimensions (movies, ratings, release dates, genres, awards, people, and roles) spread across four data sets. Cross-filtering from awards to movies to people reveals winning collaborations between top actors and directors. Selected attribute values populate a graph (bottom center) that shows individual co-occurrence relationships.

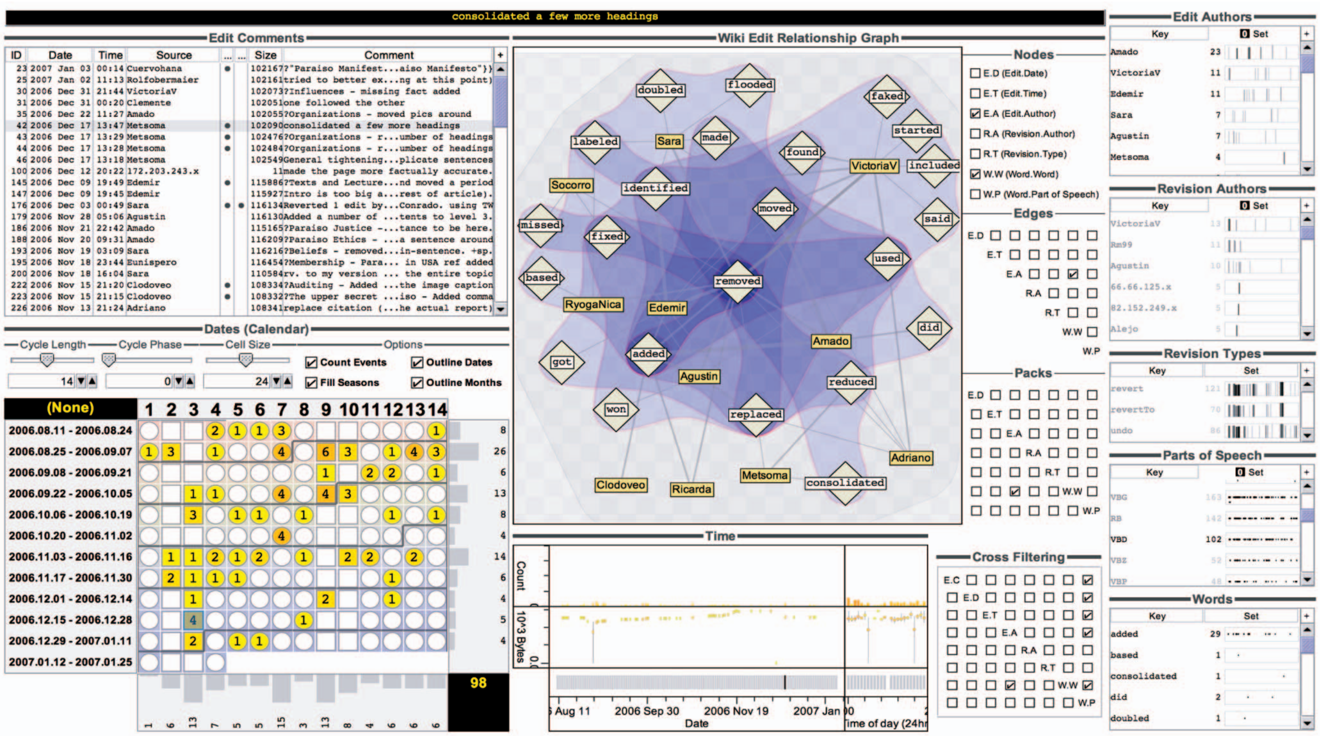

Fig. 5. Visualization of two-table, seven-dimensional data from the VAST 2008 Wiki Editors Mini Challenge, with a total of eight cross-filtering attributes. The timestamp attribute is split into two derived attributes for independent cross-filtering on date and hour of the day. A third ancillary table of keywords extracted from each edit includes part-of-speech tags as an eighth categorical dimension, used here to associate multiple edit authors with past tense verbs.

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

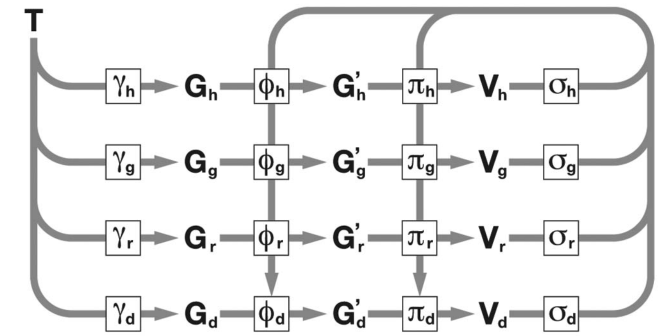

Fig. 6. Cross-filtering queries in the hotels visualization.

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.

The analytic process of cross-filtering thus closely follows Keim's adaptation [22] of Shneiderman's information-seeking mantra [23]: "Analyze First—Show the Important—Zoom, Filter and Analyze Further—Details on Demand." Cross-filtered visualization can be thought of as a focus+context technique in which all views together constitute a multidimensional context. Successive brushing and filtering interactions effectively create an ever increasing level of focus in the user's choice of dimensions. However, cross-filtering seems to better fit a zoom-and-filter model in which the meaning of "zoom" depends on one or more data dimensions being manipulated at any given time. Selection of spatial, temporal, and other quantitative data values happens in truly zoomable views such as maps, scatter plots, and time series plots. Selection of nominal and categorical data values happens in table views in which "zooming" is a contraction of the overall scrolling space. The explicit extensions in Keim's Visual Analysis Mantra parallel the primary contributions of cross-filtering to visual analysis as a process: make sure the analyst can always see what is important (attribute value selections and filtering dependencies), and provide the analyst with the means to analyze ever further (through a reversible and flexible sequence of brushing and switching interactions).

4.2 Queries

Cross-filtered visualizations build upon data transformation graphs that connect views, brushes, and switches. Each view performs four transformations on the input data:

1. grouping (γ) of 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; and
4. selection (σ) of values/sets corresponding to brushed glyphs in the view.

Each projection uses the selection in its 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, Fig. 6 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 projection 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 (Fig. 7), and as such may have a practically unlimited number of distinct,

		KEDS	Hotels	Boat	Cinegraph	Wiki
Attributes	**Nominal**	event code	guest name	vessel, ship, resolution, passenger	movie, genre, oscar, person, role	author, revision, *word*, *part of speech*
	Temporal	date (event)	date (visit)	date (encounter)	date (release)	date, time of day (edit)
	Spatial	region (countries)	location (hotel, residence)	location (encounter)	-	-
	Numerical	cooperative/conflictual weight	-	passengers, fatalities	box office, ratings, rating average	edit size (in KB)
Auxiliary Views	**Pre-filter**	list (data sources)	-	-	sliders (ratings & roles thresholds)	-
	Post-filter	map (world)	map (Pennsylvania)	map (Florida), rich drill-down table	attribute relationship graph	attribute relationship graph
	Detail	drill-down table, split time series	drill-down table	-	movie viewer	drill-down table
	Nested	scatter plot (date vs. weight)	1-D heatmap (visit count by date)	timeseries (encounters)	histogram (rating distribution)	timeseries (edit size)

Fig. 7. Variation in data attributes and auxiliary views in the five example cross-filtering visualizations.

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 plots, 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 [24] can allow 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 (Fig. 8) varies from the standard method in two major ways. First, we dealt with interactive performance limitations due to large data size by adding two prefiltering sliders. The sliders 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. Prefiltering is specified using query parameters that may be hidden and preset, or exposed and adjustable as additional controls in the dynamic queries style. Cinegraph is in this way a descendant of the FilmFinder [25].

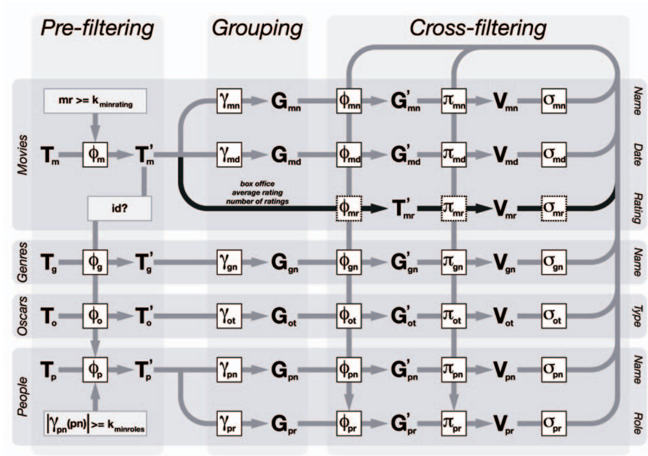

Fig. 8. Cross-filtering in the Cinegraph visualization. Input to the grouping stage consists of four tables, prefiltered to include only high-rated movies and frequent actors. The ratings "dimension" is a 3D space of movie statistics displayed in a scatter plot matrix with rubberband brushing.

Second, 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-table 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). A co-occurrence exists if the primary key of an attribute value's record, as a key into the other attribute's index, produces a non-null value upon lookup. Filtering happens in the indexes themselves, limiting valid keys to those having currently selected values only if the corresponding cross-filtering switch is activated.

We have further adapted cross-filtering for the foreign-foreign key relationships that arise in complex multiple table schemas. The semantics of co-occurrence in such cases, while often subtle, is analytically both interesting and important. For instance, cross-filtering people on roles in earlier versions of Cinegraph was counterintuitive because it calculated 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. The form of cross-column indexing determines the semantics of co-occurrence, thereby constraining which question-to-query data interrogation pathways are available to the analyst. Surprisingly, a simple pair of key-value indexes (e.g., hashtables) between foreign columns is sufficient to test whether an attribute value should be filtered under the corresponding co-occurrence relation. This works because co-occurrence is a binary relation between sets of unique attribute values, effectively turning every attribute into a primary key for a set of two-column tables.

Experience with many data sets from a variety of knowledge domains has taught that it is prudent to consider carefully the syntactic type, type semantics, external (domain) semantics, and sometimes even value distribution of all attributes when designing cross-filtered visualizations. Extensive conversation with the primary providers/users of data sets is particularly helpful for understanding nuances in the external semantics of attributes, individually and in combination, in order to capture those nuances in visual representations that succeed in supporting domain-specific exploratory and analytic needs. Seemingly innocuous data transformations—removing data by prefiltering, categorizing data by grouping values of derived attributes, and portraying data values through visual encoding—can have a profound impact on whether visual analysis tools successfully integrate into the accepted analysis practices of the domain in question, not to mention the personal analysis habits of individual users.

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.

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 3D 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 2D projections. (Although the rubberbands look and act like a 3D rectangular brush in XmdvTool, switchable filtering gives 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 (as in last-first-middle name concatenation in the hotels visualization), clustering of numerical values in high-cardinality dimensions (like rounding of 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 [26] 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 (e.g., state, county), temporal (e.g., year, month, day), 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 five 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, analysts could ask questions in the migrant boat visualization about weekend encounters in the Winter months of 2006, using Day-of-Week, Month, and Year views to select sets of derived attribute values, sets that constitute ad hoc temporal categories in the user's imagination.

5 Evaluation

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 [27]. 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, refugee migration, movies, and wikis) and 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 [5] and contributing to a doctoral dissertation in historical geography [28]. 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 [29] to provide qualitative feedback about the hotels visualization, in response to both prescribed analysis tasks and free-form exploration involving cross-filtering (see [5] for details). We have also received considerable anecdotal feedback on numerous visualizations that employ cross-filtering, such as through classroom tutorials and participation in the VAST 2008 Grand Challenge.

We experienced three 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 "selection occlusion" effect that occurs when selected items both cross-filter other views and are themselves cross-filtered out. Interactively specified sets of attribute values have real meaning in the analyst's mind, whether as a group of associated persons, a sequence of key dates, or a distribution of locations. Consequently, the cross-filtering approach strictly avoids automatic resetting, partial or otherwise, of selections of attribute values. Cross-filtering instead relies on idempotency of manual selections over a chain of user interactions. It would be undesirable, for instance, for a single switch to set off a multiview cascade of irreversible selection-filtration set intersections until a fixed point or null set were reached, destroying the analyst's ad hoc attribute groupings along the way.

Selection occlusion happens when the analyst's own interactions cause unintended visual damage to these ad hoc groupings by making some or all selected values invisible. A tendency of analysts to shift from attribute to attribute rapidly and unpredictably—especially when engaged in opportunistic exploration—makes this a frequent yet pernicious effect. One common cause is the addition of an "upstream" filter in which attribute B filters C, then A filters B, such as in the example from the migrant boats visualization shown in Fig. 9. (Occlusion of selected items during local navigation of a view coordinate space can also affect analytic utility [30]. Manual navigation occlusion is common in cross-filtered views, such as when scrolling over rows in a table view.)

Selection occlusion is exacerbated by the very idiosyncrasy of co-occurrence that makes cross-filtering so analytically useful. Filtering involves a selection of none, one, some, or all values of one attribute into a discovered set of none, one, some, or all values of another attribute. Conversely, one-step-upstream filtering visually "downgrades" a selection to none, one, some, or all of the previously selected values. Selection downgrading depends sensitively both on the values that happen to occur in the full data set as well as the potentially complex sequence of prior selection and switch interactions performed by an analyst. As a result, the selection occlusion effect is less a matter of forgetfulness on the part of the user, than one of visual indeterminacy between any given directed pair of attribute views in a longer filtering sequence.

Third, expert and student feedback, while generally positive, has repeatedly revealed concerns about the ability to see and remember more than the most recent states in the analysis process. This "out of sight, out of mind" effect is a major shortcoming of the cross-filtering approach. Anecdotally, we have found that many users have trouble executing and remembering sequences of more than two filters (A to B to C), despite the relative ease and speed—typically about 20 minutes—that it takes to train analysts how to perform the single filter interaction sequence (A to B). Longer sequences and more complex query constructions (i.e., trees and graphs of filters) are clearly useful for answering highly specific or specialized questions, yet are beyond the capabilities of most users. Moreover, users have trouble keeping track of their own questions and queries over longer analysis sessions, regardless of the complexity of queries involved. This suggests that cross-filtering by itself is good for foraging but poor for sensemaking.

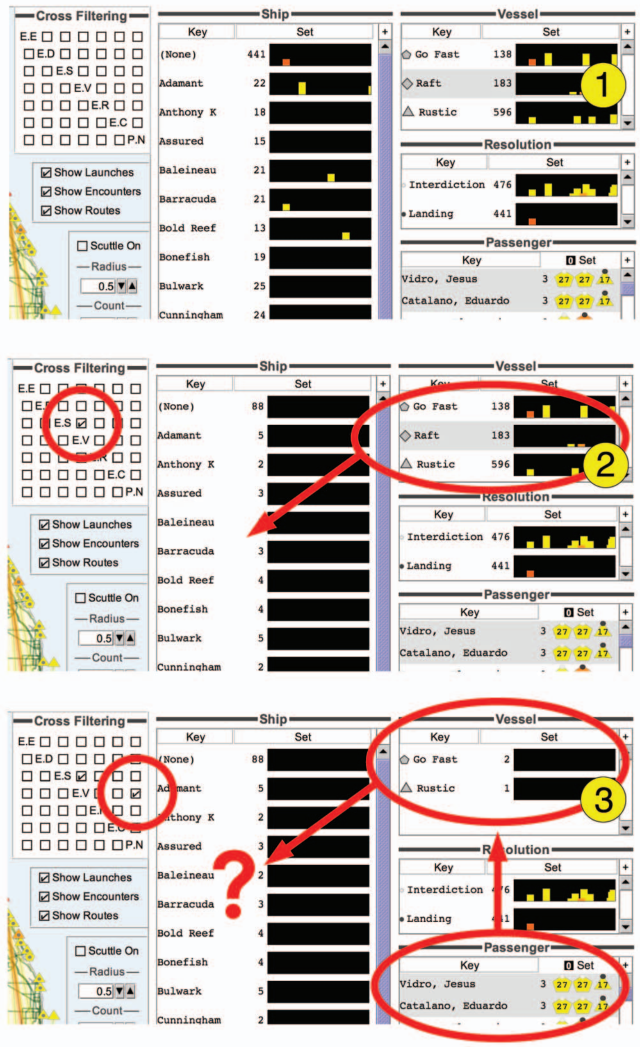

Fig. 9. Selection occlusion in the migrant boats visualization. The (1) selected vessel type that is (2) used to filter the list of ships is (3) itself hidden when filtered on selected passengers. Association between vessel types and ships is indeterminate in this visual state.

6 Discussion and Future Work

These problems have prompted suggestions for functionality to help guide cross-filtering interaction by capturing and visualizing steps in the analysis process itself. Although Improvise implements both low-level event metavisualization [31] and restorable snapshots of visualization states, neither is appropriate for the abstract, complex, discrete, yet not too frequent interactions that happen during cross-filtering sessions. One possibility is an automatically generated query-to-question user interface that translates interactive states into a visual log of formatted text fragments accompanied by restorable snapshots with timestamps and user annotations. In contrast with efforts to create natural language database query interfaces (such as the classic REMIT system [32]), this approach would attack the easier problem of expressing query results in natural language. The simple conjunctive visual form of cross-filtering queries

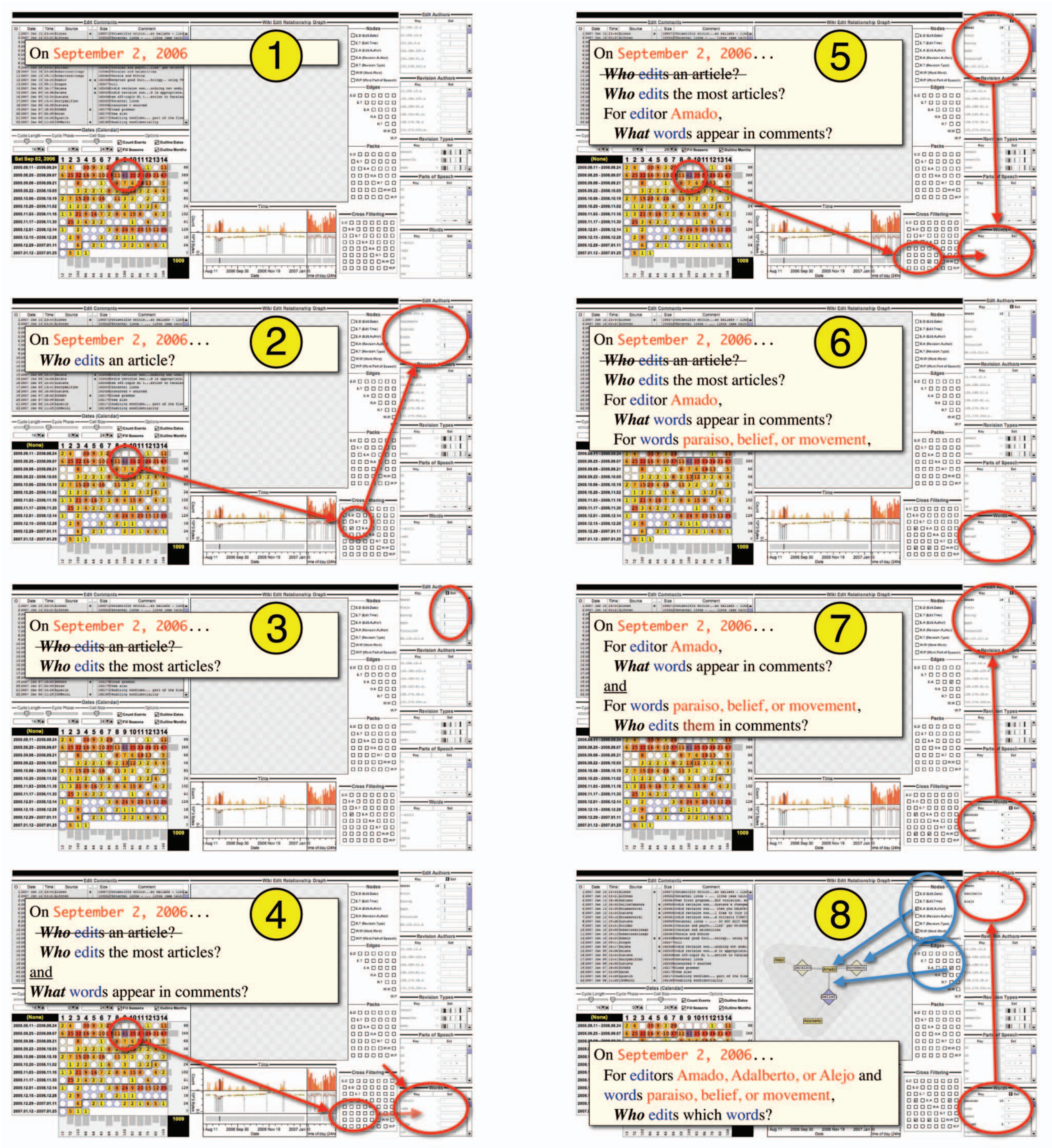

Fig. 10. Eight snapshots of a conjectural query-to-question session in the wiki visualization. The analyst: (1) picks a date; (2) filters authors on date; (3) sorts authors on edit count; (4) moves to the unfiltered table of words; (5) filters words on authors; (6) selects three words; (7) filters authors on words; and (8) associates authors with words in an attribute relationship graph. Attribute type semantics drive formatted grammar. Selected values are syntactically colored in red.

could make it relatively straightforward to translate visualization states into formatted text fragments (Fig. 10). Such a translation would need to account for the semantics of attribute types, e.g., mapping the nominal wiki editor attribute into a "who" question with an "edit" verb. Specification of such mappings could happen in a Tableau-like "grammatical encoding" builder. It would also be necessary to handle arbitrarily long lists of attribute values using either grammatical (e.g., ellipsis) or graphical (e.g., Sparkline [33]) placeholders.

We are also exploring techniques that employ compound brushing [34] to preserve visual context by replacing or supplementing filtering with multivariate "cross-highlighting" of items. It turns out to be straightforward to replace filtering with highlighting (or other visual encoding) in the cross-filtering pattern. For instance, we used modified queries from the hotels visualization to visualize data from the VAST 2008 Evacuation Traces Mini Challenge. A combination of cross-highlighting and adjustable rate animation enables visual identification and exploration of complex motion patterns, with the goal of supporting analysis of individual and group movement behaviors [35]. Visual linking of brushed items across views [36] might also be used to indicate co-occurrences (with or without filtering).

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 five 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 reduce the number of movies and people to achieve a minimum level of interactivity. Nevertheless, it is relatively easy to integrate auxiliary data sets that enhance analysis by providing additional dimensions indexed on primary data attribute values. Map layers are a common example of this dimensional extensibility.

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 (<100 ms) interactivity with upward of 100,000 data attribute values (rows times columns) on typical current desktop hardware (2 GHz dual core processor, 2 GB memory, 128 MB 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. For instance, the KEDS visualization has approximately 1 second response time when operating on its full set of nearly a million attribute values.

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 (Fig. 11) in which each row determines filtering of an attribute and each column determines filtering by an attribute. A matrix serves as a compact legend for cross-filtering in its visualization, enumerating attributes along the diagonal, and indicating active filters. Attribute labels can be color coded and include information about type or source table (as in Figs. 3, 4, and 5).

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

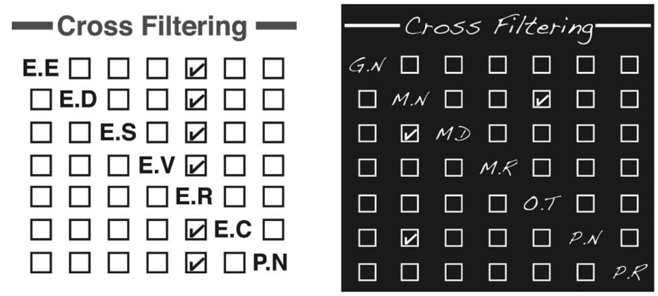

Fig. 11. Switch matrices from the visualizations in Figs. 3 and 4. In the right-hand matrix, movie names filter release dates and person names and are filtered by award types.

analysis. Implementation in Improvise has a steep and long learning curve and generally requires extensive SQL or equivalent programming experience as well as graphical design skill. Given this training, Improvise enables designers to discover new combinations of visualization components and modify existing ones through interactive exploratory design, quite unlike other toolkits and systems. We are currently looking 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.

The goal of populating a space of interactive 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. In particular, we have begun collaborating with members of the Jigsaw development team in order to better understand the strengths and weaknesses of visual analysis in both approaches with an eye toward cross-fertilization of capabilities.

7 Conclusion

Cross-filtering is a method and design pattern for fast and flexible interactive visual drill-down into fine-grained 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—at least for relational data having a modest number of mostly nominal dimensions. We have also identified key interactive usability and analytic utility limitations of the pattern. By discovering and understanding designs like cross-filtering, we hope to provide analysts with means to seek out and dissect subtle patterns in multidimensional data spaces. Moreover, we hope to develop design principles that are generalizable to different data sources, extendable to multiple interrelated data sets, and rapidly combinable and customizable into functionally rich visual tools for far-reaching visual data analysis.

ACKNOWLEDGMENTS

The author would like to thank Alan MacEachren, Anthony Robinson, and the many others at Penn State University and elsewhere who contributed directly to or commented on the visualizations in this paper, all of which were developed at the North-East Visualization and Analytics Center. This is an extended version of a paper that appeared at the IEEE Symposium on Visual Analytics Science and Technology (VAST) held on 21-23 October 2008 in Columbus, Ohio [37].

REFERENCES

[1] *Illuminating the Path: The Research and Development Agenda for Visual Analytics,* J.J. Thomas and K.A. Cook, eds. IEEE CS., Aug. 2005.
[2] P.A. Schrodt, *Event Data in Foreign Policy Analysis,* pp. 145-166. Prentice-Hall, 1994.
[3] C.E. Weaver, "Improvise: A User Interface for Interactive Construction of Highly-Coordinated Visualizations," PhD dissertation, Univ. of Wisconsin-Madison, June 2006.
[4] C. McClelland, "World Event/Interaction Survey (WEIS) Project, 1966-1978 [Computer File]," Third ICPSR ed. Univ. of Southern California, Inter-Univ. Consortium for Political and Social Research, 1999.
[5] C. Weaver, D. Fyfe, A. Robinson, D.W. Holdsworth, D.J. Peuquet, and A.M. MacEachren, "Visual Exploration and Analysis of Historic Hotel Visits," *Information Visualization,* vol. 6, no. 1, pp. 89-103, Feb. 2007.
[6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software,* first ed. Addison-Wesley Professional, Oct. 1994.
[7] G. Krasner and S. Pope, "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80," *J. Object-Oriented Programming,* vol. 1, no. 3, pp. 26-49, Aug. 1988.
[8] J. Heer and M. Agrawala, "Software Design Patterns for Information Visualization," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 853-860, Sept./Oct. 2006.
[9] E.H. hsin Chi and J.T. Riedl, "An Operator Interaction Framework for Visualization Systems," *Proc. IEEE Symp. Information Visualization,* pp. 63-70, Oct. 1998.
[10] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic Queries for Information Exploration: An Implementation and Evaluation," *Proc. Conf. Human Factors in Computing Systems (CHI),* pp. 619-626, May 1992.
[11] C. North and B. Shneiderman, "A Taxonomy of Multiple Window Coordinations," Technical Report CS-TR-3854, Dept. of Computer Science, Univ. of Maryland, 1997.
[12] C. Weaver, "Patterns of Coordination in Improvise Visualizations," *Proc. Conf. Visualization and Data Analysis,* pp. 1-12, Jan. 2007.
[13] M.O. Ward, "XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data," *Proc. IEEE Conf. Visualization,* pp. 326-333, 1994.
[14] Y.-H. Fua, M.O. Ward, and E.A. Rundensteiner, "Structure-Based Brushes: A Mechanism for Navigating Hierarchically Organized Data and Information Spaces," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 2, pp. 150-159, Apr.-June 2000.
[15] C. Weaver, "Infovis 2007 Contest Entry: Cinegraph," *Proc. IEEE Symp. Information Visualization (Compendium),* Oct. 2007.
[16] C. Ahlberg, "Spotfire: An Information Exploration Environment," *ACM SIGMOD Record,* vol. 25, no. 4, pp. 25-29, Dec. 1996.
[17] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: A System for Query, Analysis, and Visualization of Multi-Dimensional Relational Databases," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 1, pp. 52-65, Jan. 2002.
[18] J.D. Mackinlay, P. Hanrahan, and C. Stolte, "Show Me: Automatic Presentation for Visual Analysis," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1137-1144, Nov./Dec. 2007.
[19] J. Stasko, C. Görg, Z. Liu, and K. Singhal, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 131-138, Oct. 2007.
[20] *QlikView,* http://www.qlikview.com/, Feb. 2009.
[21] 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," *J. Information Visualization,* vol. 1, nos. 3/4, pp. 211-228, Dec. 2002.
[22] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in Visual Data Analysis," *Proc. Int'l Conf. Information Visualization (IV),* pp. 9-16, July 2006.
[23] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," *Proc. IEEE Symp. Visual Languages,* pp. 336-343, Sept. 1996.
[24] J.C. Roberts, "Multiple-View and Multiform Visualization," *Proc. SPIE Conf. Visual Data Exploration and Analysis VII,* R. Erbacher, A. Pang, C. Wittenbrink, and J. Roberts, eds., pp. 176-185, Jan. 2000.
[25] C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays," *Proc. Conf. Human Factors in Computing Systems (CHI),* pp. 313-317, 479-480, Apr. 1994.
[26] C. Olston, A. Woodruff, A. Aiken, M. Chu, V. Ercegovac, M. Lin, M. Spalding, and M. Stonebraker, "DataSplash," *Proc. ACM SIGMOD,* pp. 550-552, June 1998.
[27] C. Plaisant, "The Challenge of Information Visualization Evaluation," *Proc. ACM Conf. Advanced Visual Interfaces (AVI),* pp. 109-116, May 2004.
[28] D. Fyfe, "Commerce and Sociability in Small-Town America: Explorations in Historical GIScience," PhD dissertation, The Pennsylvania State Univ., Jan. 2008.
[29] 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,* vol. 47, no. 2, pp. 8-21, 2005.
[30] N. Elmqvist and P. Tsigas, "A Taxonomy of 3d Occlusion Management for Visualization," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 5, pp. 1095-1109, Sept./Oct. 2008.
[31] A.C. Robinson and C. Weaver, "Re-Visualization: Interactive Visualization of the Process of Visual Analysis," *Proc. GIScience Workshop Visual Analytics & Spatial Decision Support,* Sept. 2006.
[32] B.G.T. Lowden and A.N.D. Roec, "The REMIT System for Paraphrasing Relational Query Expressions into Natural Language," *Proc. 12th Int'l Conf. Very Large Data Bases,* pp. 365-371, Aug. 1986.
[33] E. Tufte, *Beautiful Evidence.* Graphics Press, 2006.
[34] H. Chen, "Compound Brushing," *Proc. IEEE Symp. Information Visualization,* pp. 181-188, Oct. 2003.
[35] C. Weaver, "Cross-Dimensional Visual Queries for Interactive+ Animated Analysis of Movement," *Proc. GIScience Workshop GeoSpatial Visual Analytics,* Sept. 2008.
[36] C. Collins and S. Carpendale, "Vislink: Revealing Relationships amongst Visualizations," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1192-1199, Nov./Dec. 2007.
[37] C. Weaver, "Multidimensional Visual Analysis Using Cross-Filtered Views," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 163-170, Oct. 2008.

Chris Weaver received the PhD degree in computer science from the University of Wisconsin-Madison in 2006. He is currently an assistant professor in the School of Computer Science and associate director of the Center for Spatial Analysis at the University of Oklahoma. Until 2008, he was a research associate in the GeoVISTA Center in the Department of Geography at The Pennsylvania State University, where he remains affiliated as a founding member of the North-East Visualization and Analytics Center. His research interests include information visualization, visual analytics, and visual query languages. He is a member of the IEEE Computer Society.