Conjunctive Visual Forms

Chris Weaver, *Member, IEEE Computer Society*

Abstract—Visual exploration of multidimensional data is a process of isolating and extracting relationships within and between dimensions. Coordinated multiple view approaches are particularly effective for visual exploration because they support precise expression of heterogeneous multidimensional queries using simple interactions. Recent visual analytics research has made significant progress in identifying and understanding patterns of composed views and coordinations that support fast, flexible, and open-ended data exploration. What is missing is formalization of the space of expressible queries in terms of visual representation and interaction. This paper introduces the Conjunctive Visual Form model in which visual exploration consists of interactively-driven sequences of transitions between visual states that correspond to conjunctive normal forms in boolean logic. The model predicts several new and useful ways to extend the space of rapidly expressible queries through addition of simple interactive capabilities to existing compositional patterns. Two recent related visual tools offer a subset of these capabilities, providing a basis for conjecturing about such extensions.

Index Terms—Boolean query, brushing, conjunctive normal form, exploratory visualization, multiple views, visual abstraction.

◆

1 INTRODUCTION

Many information collections of interest in visual analytics [18] contain data that is predominantly nominal in nature, even when they contain significant geospatial, temporal, numerical, and/or categorical information. Geospatial and temporal dimensions often come in the form of place names and discrete dates. In the scientific, technological, and financial domains, information collections almost always go beyond the merely quantitative. Even when data sets contain quantitative dimensions, the sets of unique numerical values are often relatively small, and in some cases are even smaller than the sets of unique nominal values. Homogeneous quantitative data sets from sources such as scientific instrumentation, sensor networks, and computer simulations are huge but few in comparison to the numerous heterogeneous data sets from health care, finance, business logistics, social networks, the humanities, intelligence, and so on.

Visual analysis of multidimensional data from such sources often involves looking for multiple simultaneous patterns of association between entities across dimensions. Association in visual analysis goes beyond the colloquial sense—of relationships between people and organizations—to encompass not only spatial and temporal relationships, but also relationships across multiple scales between data values and aggregates of any quantitative or nominal type. An analyst's ability to associate data of different types depends on a means to select the data of interest, the form of representation, and the manner of interaction, in accordance with the semantics of association being investigated. Moreover, simultaneous investigation often requires multiple independent associations of a particular attribute with other attributes, such as when exploring patterns of events or groups of people with respect to both locations (over all dates) and dates (over all locations).

Visual exploration of associations is an open-ended process of looking for and examining interesting regions in an abstract data space by applying parameterized visualization transforms to generate visual results [8]. Visual identification of regions as interesting, and hence worthy of further examination, is a subjective perceptual and cognitive capacity that the analyst brings to this process. Visual specification of a region is similarly analyst-driven, involving selection of objects, points, and regular or irregular extents in a multidimensional visual abstraction. Representation of a region, on the other hand, is an objective description of a collection of selected objects and geometries in terms of individual data dimensions. Taken together, the visual exploration process relies on the analyst's ability to quickly and flexibly tag subsets of values within and between dimensions as interesting [24].

Visual analysis tools such as Jigsaw [17] and the historic hotel register visualization [23] extend this set-oriented style of visual exploration by using multiple coordinated views to support cross-dimensional drill-down capabilities. Supporting flexible visual exploration in these tools boils down to providing analysts with the ability to rapidly select unique attribute values and toggle filtering and highlighting between pairs of views in order to pose evolving sequences of complex multidimensional set queries. The cross-filtered views [22] model generalizes this approach in terms of a mesh of interwoven views, brushes, and switches/toggles. In this pattern, each cross-filtered view displays the unique values of a particular data attribute in a type-appropriate way. Each view also allows brushing to select a subset of displayed values. For each directed pair of views, an interactive switch (such as a checkbox) toggles filtering from one view to the other. The semantics of filtering is that of simple co-occurrence between data columns, namely to show only those values in view B that co-occur in any data records with the values selected in view A. As useful as cross-filtering designs are for visual exploration, their reliance on simple pairwise disjunctive filtering of brushed values means that they are functionally incomplete in terms of value set operations, and hence limited in query expressiveness. Negation, for instance, is not considered in the cross-filtering model, despite its importance in exploration and analysis for posing questions and framing answers.

The conjunctive visual form model extends the cross-filtered views model by translating the abstract representation and interaction semantics of cross-dimensional filtering designs into a system of formulas in conjunctive normal form. This formalism is *not* presented here as part of a symbolic interface to be manipulated directly by the user. Rather, it serves as a conceptual framework for visualization researchers and tool designers to reason about patterns of interaction that support rapid user expression of multidimensional set queries. Modeling interaction in terms of a logical formalism provides a basis for characterizing the capabilities and limitations of existing tools. It also leads directly to predictions of ways to radically expand the space of useful expression in these tools by adding simple new forms of interaction to views. In particular, logical negation prompts both an *include-exclude-ignore* trichotomy of brushing and a *conjunction-disjunction* dichotomy of views (as set containers). Together, the models provide understanding of how particular patterns of coordinated views can be synthesized into tools that support rapid and flexible expression of the kinds of complex visual queries needed for open-ended exploration and analysis.

2 THE CONJUNCTIVE VISUAL FORM MODEL

A conjunctive visual form (CVF) captures the representational states and interactive transitions of a cross-filtered views design as an evolving set of boolean formulas in conjunctive normal form (CNF).

● *Chris Weaver is with the School of Computer Science and the Center for Spatial Analysis at University of Oklahoma, E-mail: weaver@cs.ou.edu.*
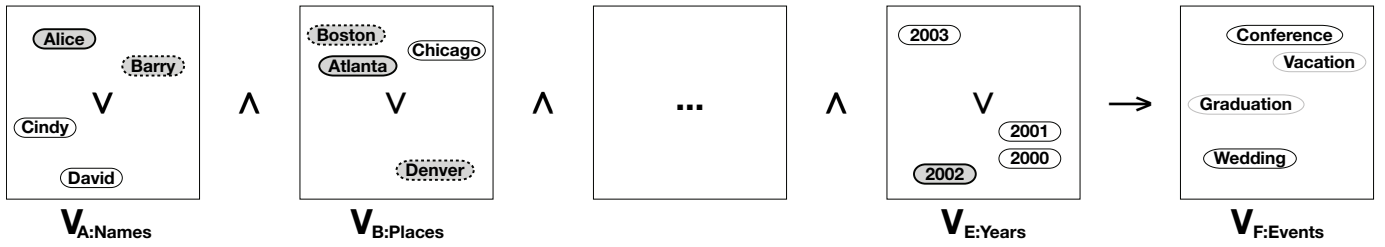
Fig. 1. Conjunctive visual form, in which a target view ($V_{F:Events}$) differentially encodes unique events that co-occur (or not) with at least one selected value from every other view. In all views, selected values are represented using darker fill color. Negated selection is indicated by dashed edges.

The three-level hierarchy of clauses in CNF—literals, disjunctions, conjunctions—maps directly onto a three-level hierarchy of interactive visualization design consisting of values, views, and coordinations. This direct map makes it possible to translate existing visual designs into CVF. It also makes it possible to extrapolate from the set of possible CNF-to-CNF transformations, not only to characterize existing representations and interactions, but also to identify new representations and interactions that could be added to visual analysis tools in ways that open up new avenues for multidimensional exploration.

In the model, each data attribute is paired with a view that visually represents unique attribute values in a type-appropriate way, such as a table view for people's names, a calendar for dates, or a map for latitude-longitude points. The analyst can tag a subset of values in each view using well-known interactions such as clicking or rubber-banding. Highlighting of values in a subset signifies that the values are of *collective interest* in terms of potential relationships to the values of other attributes, as displayed in other views.

Correspondingly, the selected subset in each view is treated as a *visual disjunction* (eq. 1). Selection of objects as visual disjunction is in keeping with the many-to-many nature of brushing, in which uniform highlighting conveys set membership but not one-to-one correspondences. The same holds for selection of spatial regions by navigation, in which region-based containment/overlap/"hits" of objects conveys mutual membership in an ad hoc set expressed by the analyst during exploration. A pattern observed in the subset may in fact only involve an even smaller subset. Similarly, observation of a relationship within another data dimension may only hold for a subset of selected items, and perhaps even only for individual items.

$$V_A \equiv (a_0 \vee a_1 \vee \cdots \vee a_{n-1})$$
$$V_B \equiv (b_0 \vee b_1 \vee \cdots \vee b_{m-1}) \quad (1)$$

Each literal clause in a visual disjunction possesses a truth value just as in other systems of logic. Evaluation to either *true* or *false* is determined by the underlying context of interaction used for selection of objects or space. In the case of normal brushing of unique attribute values, a literal is a test of whether a particular value of an attribute is currently selected, e.g. $a_0 \equiv (a ==\text{`Alice'})$. For persistent, one-dimensional rubberbanding of values, a literal tests range containment, e.g. $e_0 \equiv (r_{min} <= e <= r_{max})$. Other forms of interactive selection produce similar literal clauses. Fuzzy matching and text searching can be incorporated using clauses like $f_0 \equiv (f \; \texttt{startswith} \; \text{`Conf'})$.

All literal clauses are resolved from their functional form upon addition to a disjunctive clause, in essence treating literals as interactively idempotent by inlining the evaluated truth value into the disjunction. This corresponds to the common interactive semantics of selections in which items in a view remain selected even after the geometry of the selection gesture—or, say, the text input used for a string search—has been forgotten. In this way, CVF follows boolean logic except in the way that truth values of literals are determined.

Just as a formula in conjunctive normal form is a conjunction of disjunctive clauses, a conjunctive visual form is a conjunction of visual disjunctions. A *conjunctive visual form* is thus a selected set of views that collectively specify a compound interactive co-occurrence test in which each unique attribute value of a given dimension passes if and only if, for every view, that value co-occurs in at least one data record with at least one selected unique attribute value in that view. The view for that dimension evaluates the test to differentially encode its attribute values. For example, figure 1 depicts a conjunctive visual form for the data set in table 1. The target view ($V_{F:Events}$) highlights two events ('Conference', 'Wedding') for which: (1) one person ('Alice') was involved or another ('Barry') was not; one location ('Atlanta') was involved or either of two locations ('Boston', 'Denver') were not; and (3) the event took place in a particular year (2002).

As this example suggests, an essential element of expressiveness under the CVF model is the ability to negate literal clauses, paralleling the logical negation of literals under CNF, e.g.:

$$(a_0 \vee \neg a_1) \wedge (b_0 \vee \neg b_1 \vee \neg b_3) \wedge \cdots \wedge (e_2) \quad \rightarrow \quad (f_0 \vee f_3) \quad (2)$$

Unlike CNF, however, CVF allows negation not only of the literal clauses within visual disjunctions, but also the disjunctions themselves (in a carefully prescribed manner). As it turns out, this one relaxation of CNF opens the door to a simple interaction that substantially increases exploratory query expressiveness.

A basic CVF visualization consists of a set of views, each paired with a data attribute. Each view is cross-encoded using a conjunction of the visual disjunctions of the other views. This nearly symmetric set of interdependencies makes it possible to drill-down into multidimensional data from any direction, switching from attribute to attribute and selecting increasingly specific subsets of values of those attributes:

$$\begin{aligned}
V_B \wedge V_C \wedge \cdots \wedge V_F &\rightarrow V_A \\
V_A \wedge V_C \wedge \cdots \wedge V_F &\rightarrow V_B \\
V_A \wedge V_B \wedge \cdots \wedge V_F &\rightarrow V_C \\
&\vdots \\
V_A \wedge V_B \wedge \cdots \wedge V_F &\rightarrow V_E \\
V_A \wedge V_B \wedge \cdots \wedge V_E &\rightarrow V_F \quad (3)
\end{aligned}$$

Visual exploration under the model consists of interactively-driven transformations of conjunctive visual forms. These transformations are inspired by the various normal form-preserving logical transformations that can be applied to conjunctive normal forms. In particular, the three-level hierarchy of clauses in CNF constrains modifications of individual formulas to combinations of additions, removals, and negations of conjunctions, disjunctions, and literals.
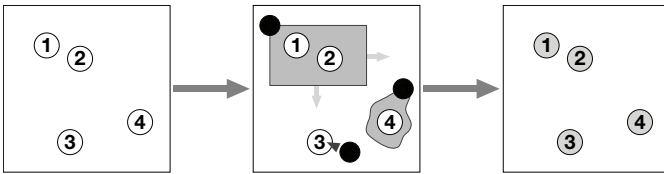
A: Name	B: Place	...	E: Year	F: Event
Alice	Atlanta		2002	Conference
Alice	Atlanta		2002	Wedding
Alice	Boston		2000	Vacation
Barry	Boston		2000	Graduation
Barry	Chicago		2001	Vacation
Cindy	Atlanta		2001	Graduation
Cindy	Atlanta		2003	Wedding
David	Denver		2002	Conference

Table 1. Example data set used in figures 1, 2, 3, 9, and 10.

Fig. 2. Selecting literals. Brushing (σ) using click (3), rubberbanding (1,2), or lasso (4) interactions to include-select a set of values.
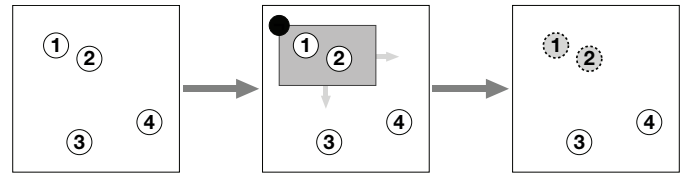

Fig. 3. Negating literals. Negative brushing ($\neg$) using a modal rubberbanding interaction to exclude-select a set of values.

Combinations of modifications can be arbitrarily complex. One can imagine individual interactions that trigger different degrees of visual transformation, ranging from simple addition/removal/negation of a single value by brushing, to extensive rearrangement of selections and filters across multiple views. However, the difficulty of predicting usefulness and usability increases with transformation complexity. For instance, negation of conjunctions or disjunctions under CNF requires renormalization to retain conjunctive normal form. What would it mean to "renormalize" a CVF visualization? The sharing of disjunctions by each view's conjunction means that even a single simple interactive modification may cause multiple complex visual effects.

The cross-filtered view approach demonstrated that analysts can effectively pursue visual exploration of mixed multidimensional data using sequences of simple interactions. We proceed here to look at whether visual exploration under the conjunctive visual form model can be similarly supported using only sequences of simple interactions that perform "atomic" visual manipulation of clauses. The following subsections consider ten potential types of modification/manipulation and their ramifications for visual exploration.

2.1 Selecting Literals

Collection of things into groups is an essential operation of visual exploration. In visualization, collection is simply a matter of identifying and marking a set of visually encoded objects as a coherent whole within a spatial context (i.e. the coordinate space of a view). This process is generally referred to as selection, consisting of: (1) an interaction that targets objects or a region that contains objects; and (2) a visual differentiation of selected versus unselected items, imposed during or after the interaction. Brushing plus highlighting is a common form of selection [1]. Here we distinguish "brushing" in a view in order to highlight items in that view, from "brushing-and-linking" in a view to highlight associated items in another view (as well as possibly, but not necessarily, the brushed view itself).

Selecting literals under the model is nothing more than selection of visualized objects in the normal sense (figure 2). Building and modifying a visual disjunction thus follows the common ways in which analysts can select and deselect objects in views. Multiple disjunctions can be modified quickly because each view supports independent interactive collection of zero or more unique values of its attribute through selection. Moreover, the interaction(s) and representation(s) used for selection can be tailored to each attribute's type and semantics on a view-by-view basis [25].

2.2 Negating Literals

Just as selection allows the analyst to express queries about included data values and their specific co-occurrences, literal negation makes it possible to express queries about excluded values and their lack of specific co-occurrences, e.g. that 'Barry' was not involved in either of two events. The question becomes one of how to visually negate literal clauses. In the formulation in eq. 1, interaction with and representation of literals amounts to a single "channel" of brushing and highlighting, in which interaction and representation distinguishes included literals from ignored literals in terms of selected versus unselected values.

Negation of literals thus requires a third class of selection state in order to express queries that mean "these, but not those, and ignore all the rest". Views in conjunctive visual forms therefore must provide interaction and representation for negative (negated) selection distinct from those for positive (normal) selection, thereby supporting three selection states: *included*, *excluded*, and *ignored*. Three-state selection may be a simple matter of modes, such as using a negating modifier key with existing selection interactions (figure 3). The challenge may be in combining visual channels into an effective single visual encoding of value plus selection state [20, 2]—adding positive or negative meaning on top of the basic neutral/ignored visual encoding of values—and to do so for each attribute type within its view context.

2.3 Rearranging Literals

Under normal selection, the set of selected values in a view can range from the empty set to all values. The corresponding disjunction can thus consist of any combination of literals, including none at all. Extending selection to three-states—include-selected, exclude-selected, unselected—means that any given literal may be either present, present and negated, or absent from the disjunction. There are thus six total classes of modification involving the simple transfer of a subset of values in one selection state into another, each of which has the effect of adding, removing, or negating literals in the corresponding disjunctive clause. The modifications for selecting and negating literals cover four of these classes. The remaining two, polarity-reversal modifications, express "not these" (or "not not these"!) by moving indicated values directly from one selected state to the other (such as to look for different dates on which 'Barry' was involved rather than not involved).

Although one can imagine arbitrarily complex modifications to a disjunction involving transfers between two or even all three selection states, in general these fall outside the scope of "atomic" manipulations. A special case worth noting is that of complement modification, in which literals in a mixed set are each transferred to their complementary set; this would be useful, for instance, to switch to "everyone else" within an associated group of people. Like simple
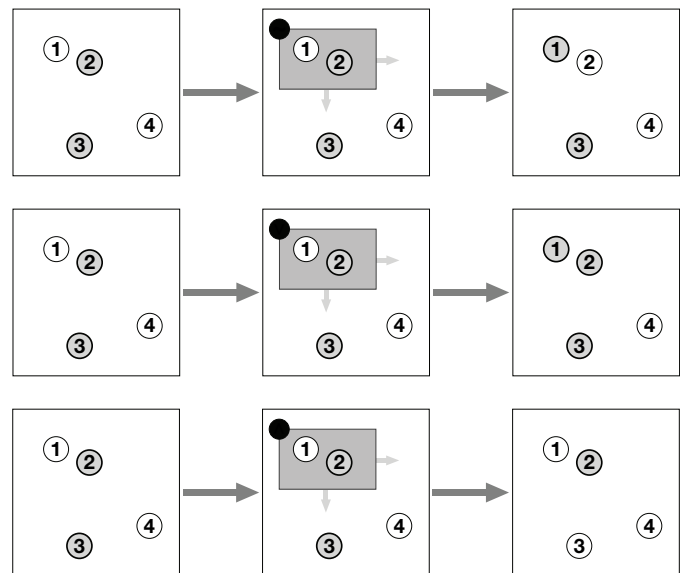

Fig. 4. Rearranging literals in a disjunction. Modified brushing interactions select the complement ($\sim$, top), set disjunction ($\vee$, middle), or set conjunction ($\wedge$, bottom) of brushed values relative to unbrushed ones.
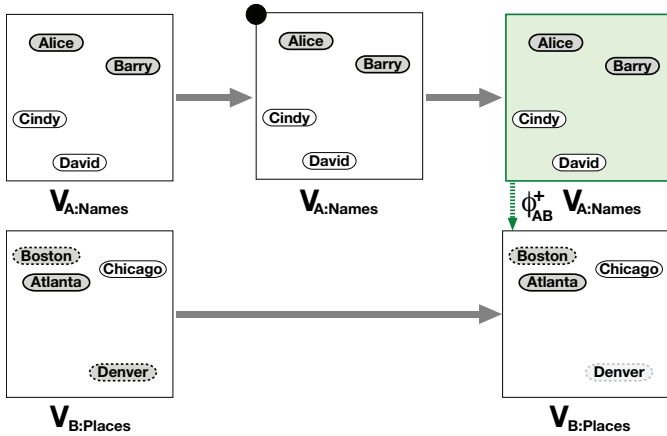
Fig. 5. Selecting a disjunction. A positive filter (Φ^+_{AB}) on V_B elides places that fail/pass co-occurrence with include/exclude-selected names in V_A.

Fig. 6. Negating a disjunction. A negative filter (Φ^-_{AB}) on V_B elides places that pass/fail co-occurrence with include/exclude-selected names in V_A.

transfers, there are multiple classes of complementing, one for each directed pair of selection states. Composite brushing [11] could be similarly extended to perform AND, OR, XOR, and NOT set operations on mixed-selection values of the same or different polarities. Figure 4 depicts a few of the simpler possibilities. The clear challenge here is to compose all of the possible modes of modification into a usable collection of inclusive and exclusive grouping interactions that the analyst can choose and execute quickly during visual exploration.

2.4   Selecting Disjunctions

Visualization tools such as ManyEyes [19], Tableau/Show Me [10], and Jigsaw provide the highly useful capability to interactively select which dimensions are of exploratory interest. Although the overall balance of *flexibility of dimension selection* versus *homogeneity of visual abstraction over those dimensions* varies widely between these tools—arguably shifting toward more flexibility but also increased homogeneity in the order given—dimension selection is clearly essential for open-ended visual exploration in general.

The modifications and corresponding interactions to this point have focused on literal clauses within individual disjunctions. Moving up to the next level of the CNF clause hierarchy shifts the focus to disjunctive clauses within individual conjunctions. Because each disjunction represents the collective selection state of a view, composing a set of disjunctions corresponds to building a multidimensional query by selecting a set of dimensions. This necessitates a means to interactively choose which views to include in each conjunction. For individual conjunctions, this can be done by applying an operation to views themselves, with a corresponding change in overall appearance (figure 5).

Selecting sets of disjunctions for multiple views' conjunctive forms, however, requires a many-to-many interface. For a basic CVF visualization using cross-filtering (eq. 3), this need can be addressed for N-dimensional data using an N-by-N matrix of checkboxes to toggle cross-filtering between any directed pair of views (with one view per dimension). Other visualizations for selecting dimensional pairs might include graphs with dimensions as nodes and brushable directed edges as filter toggles, or scatter plot matrices in which the scatter plots themselves are also independently triggerable filter toggles (similar to navigation in [4]). An ability to filter and drill-down into the dimensions themselves—such as by reusing cross-filtered CVFs for visual exploration of metadata—could be used to extend visual exploration to very high-dimensional heterogeneous data sets from interconnected sources. Data and metadata respectively map to the disjunctive and conjunctive levels of the clause hierarchy in a clean, separable way.

2.5   Negating Disjunctions

In evaluating cross-filtered visualizations (and using them ourselves), we have often observed a desire on the part of the analyst to be able to quickly express negation of different dimensional types using both
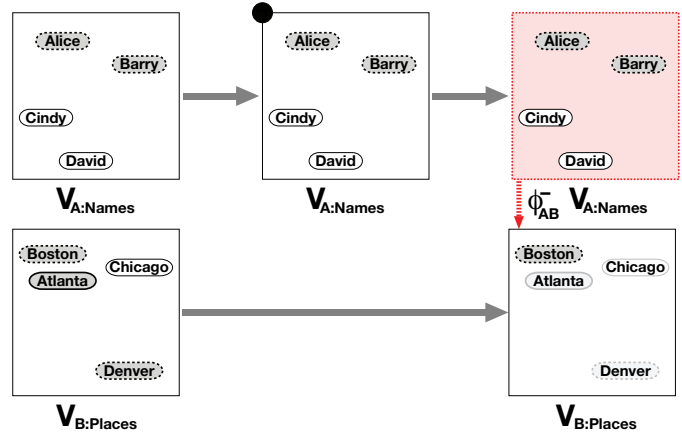
universal qualifiers ("nowhere", "never", "none") and existential qualifiers ("not here", "not now", "not this").

Such relations are impractical to express under the cross-filtered views model. Although literal negation is an improvement, it involves modification of disjunctive forms in ways perhaps not easy to reverse. (Selection reuse [6] could help in this regard, and be extraordinarily useful in basic cross-filtering as well as extensions using CVF.) It would be faster and reversible to apply a simple NOT operation to an entire view; that is, to tests of co-occurrence with the values it displays. This is possible to do by relaxing CNF to allow negation of disjunctive clauses. As with selecting disjunctions, there needs to be a way to apply and represent a persistent negation operation over a view as a whole (figure 6). Similarly, any many-to-many interface (like the N-by-N cross-filtering matrix) would need to be extended to differentiate negation of disjunctions/views from other logical states.

An unanticipated benefit of disjunction negation is that it enables *visual conjunction* as well as *visual disjunction* in cross-filtering queries. In boolean logic, De Morgan's laws can be used to transform a negated disjunction to and from a conjunction of negations:

$$\neg(a_0 \vee a_1 \vee \cdots \vee a_{n-1}) \Leftrightarrow (\neg a_0 \wedge \neg a_1 \wedge \cdots \wedge \neg a_{n-1}) \qquad (4)$$

This transformation means that it is possible to specify conjunctive co-occurrence relations by combining literal negation with disjunction negation. Queries like "show me places where all of these people appear" are visually expressed as "show me places where none (not any) of these people fail to appear". The challenge here is to provide interactions and representations for visual conjunction that: (1) express the latter kind of query, (2) can be interpreted immediately and easily as the former kind of query, and (3) do not conflict with the interactions and representations already being used for visual disjunction. Interestingly, the practical duality of positive versus negative highlighting of both values (literals) and views (disjunctions) suggests that this is possible even for heterogeneous data types and visual encodings. Discerning polarity might occur through perception of the encoding (preattentively?) rather than cognition of the logical form.

2.6   Rearranging Disjunctions

Visual exploration of metadata, above and beyond mere selection of dimensions/views, could greatly facilitate visual exploration of high-dimensional data. Rearranging disjunctions in conjunctions could thus be as valuable as rearranging literals in disjunctions. Rearrangement could even be accomplished in much the same way, using additional interactions and representations to perform AND, OR, XOR, NOT, and complement operations on sets of dimension/view "values".

A critical detail of disjunction rearrangement is the difference in co-occurrence semantics between *absent* disjunctions and *empty* disjunctions. An absent disjunction means that filtering on that dimension is inactive, and thus has no effect on co-occurrence calculations.
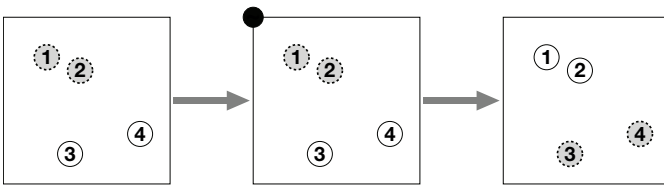
Fig. 7. Complementing a disjunction. Applying a complement operator (∼) to a view inverts two or more selection states of all attribute values.

An empty disjunction means that conjunctions test to see if values co-occur with empty include- and exclude-selection sets. This test is always *true* (by convention) unless that disjunctive clause is negated. Moreover, at any given time, the analyst may choose to explore using combinations of filter toggling and select-all/deselect-all interactions. It is therefore important to convey query semantics at the conjunctive level carefully regardless of the interface used to rearrange disjunctions, especially when the set of dimensions is fixed (and thus might be expected to conform exactly to eq. 3).

2.7 Complementing Disjunctions

In most view implementations, brushing interactions are limited to single item selection by clicking glyphs or multiple item selection by rubberbanding or lassoing. Many views do support additive brushing and a "select all" keyboard command. Some views support subtractive brushing as well; complex region specification in many image editors involves both addition and subtraction of regions.

Complementing disjunctions is a useful variation of literal rearrangement that happens at the conjunctive level, in parallel with selecting and negating of disjunctions. The operation complements literals in a view/disjunction in the same way as in literal rearrangement, but does so in a persistent and easily reversible manner (figure 7). As in complement brushing of literals, the three selection states prompt three different complement operations. A single "cyclic complement" operation—from unselected to selected to negated to unselected, for instance—could be used to reduce the number of distinct types of interaction that must be remembered, in exchange for an occasional increase in the number of actions. This strategy could be employed to combat interactive mode multiplication in other modifications as well.

2.8 Reflective Filtering

In the basic CVF formulation (eq. 3), the conjunctive normal forms are not completely symmetric. Each conjunctive clause lacks a disjunctive clause, namely the one corresponding to the form's own view. Reflective filtering adds the ability for a view to filter itself by allowing each view to include its disjunctive clause in its own conjunctive form:

$$V_A \wedge V_B \wedge V_C \wedge \cdots \wedge V_F \quad \rightarrow \quad V_A \qquad (5)$$

Reflective filtering (figure 8) enables rapid toggling of the encoding of selected values in a view, from highlighted to elided and back. This enables visual exploration of a single attribute within a view in much the same way as exploration of multiple attributes across views. The view becomes a "self-focusing" context in which sequences of selection gestures can be used to easily drill-down into a large set of attribute values as a precursor to exploration with other attributes.

A difficulty of reflective filtering is the visual asymmetry inherent in elision as compared to highlighting. Highlighting normally involves two different visible encodings of selected versus unselected state. Using full invisibility (as when filtering) to encode one of these states eliminates location and shape cues that help analysts target gestures. Moreover, because reflective filtering involves both cause (selection) and effect (highlighting/eliding) in the same visual context, visual feedback during and after brushing needs to accommodate both current and updating selection states. A reserved low opacity encoding might be used to indicate self-filtered values in the absence of other filtering. Elided values could instead adopt such an encoding only during active brushing. Composite brushing operations could be as helpful in
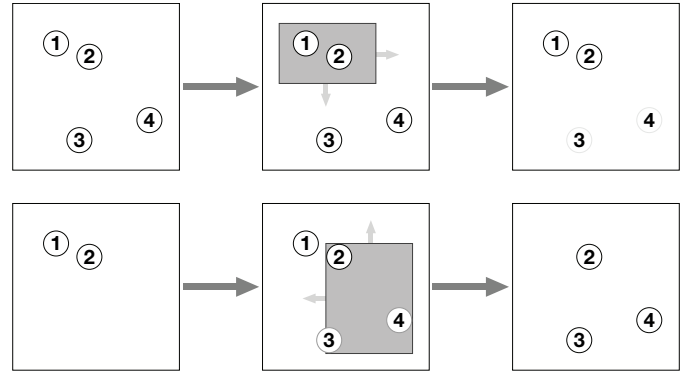

Fig. 8. Reflective filtering. A view's own attribute values are selected by normal rubberband brushing. Top: Self-unselected values are dimmed. Bottom: Self-unselected values are invisible except during brushing.

reflective filtering as in literal rearrangement, but would be complicated by the additional channels of encoding that would be necessary to indicate whether a value is reflectively filtered or not.

2.9 Perpendicular Filtering

It is often helpful to use the same disjunctive clause to filter multiple views. For instance, given a set of names, one can show the places involving those same people in a map, and also show the years involving those people in a calendar view. Selecting different names updates both map and calendar view. It is also often useful to engage in multiple paths of querying simultaneously, such as to show *where* some people were involved in certain activities, at the same time as showing *when* other people were involved in the same activities (figure 9).

In the basic CVF formulation (eq. 3), every view contributes the same disjunctive clause to the conjunctive forms of all views. For instance, V_B, V_C, V_D, V_E, and V_F all use the same clause defined by V_A. Supporting multiple associations for any particular attribute is thus a matter of allowing the conjunctive form of each view to have its own disjunctive clauses for any or all attributes. Visualizations thus need to support multiple views of an attribute, in order to allow the analyst to differentially filter views by specifying different disjunctive clauses of an attribute's values. Views of the same attribute may have different visual encodings, as in multiform views [14]. Each view maintains independent inclusive and exclusive selections over all values. Any view can be cross-filtered on any combination of views, meaning that perpendicular views for an attribute can be freely shared by other views.

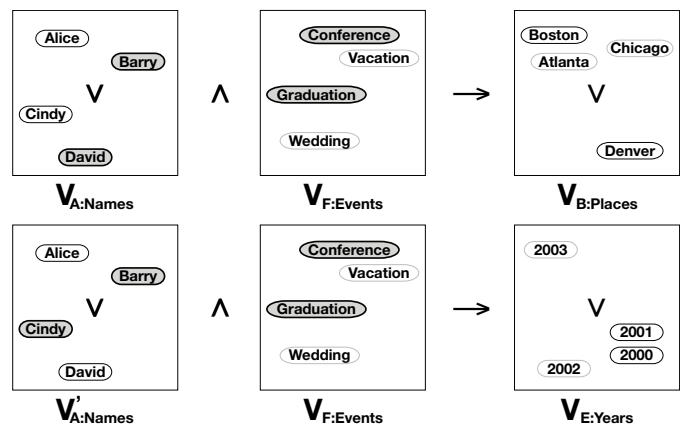Perpendicular filtering makes it possible to use different sets of

Fig. 9. Perpendicular filtering. Highlighting of places (in $V_{B:Places}$) and years (in $V_{E:Years}$) depends on the same disjunctive clause of events ($V_{F:Events}$), but different disjunctive clauses of names ($V_{A:Names}$, $V'_{A:Names}$).

attribute values for different purposes within and across multiple queries. Perpendicular views can be created, deleted, and activated for filtering—thus adding to the corresponding conjunctive forms—as needed to support evolving investigation of associations. A facility to split and merge value sets would enable diverging and converging investigation of associations, perhaps using pools of views with selection states saved/restored/minimized in a tray or sandbox [13].

2.10 Parallel Filtering

An analyst's ability to "find a needle in a stack of needles" depends on a means to specify and apply different sets of data values—whether names, locations, dates, times, or other concrete or abstract entities and measures—in a flexible, rapid, and reversible manner. Specification of any set does not happen in a vacuum, but rather through an analyst-driven process of discovering co-occurrence semantics that are explicitly associated with set members as a collective whole. A facility to create, delete, edit, swap, and view multiple sets would support comparison of current multidimensional hypotheses and reuse of past ones. This would be especially beneficial for visual analysis of complex phenomena involving potentially numerous idiosyncratic relationships within and between multiple mixed-type dimensions.

In the basic CVF formulation (eq. 3), the conjunctive form for any given view contains a single disjunctive clause for each of the other views. For instance, V_A contains a single clause for each of V_B, V_C, V_D, V_E, and V_F. With parallel filtering, conjunctions can be extended to allow multiple, independent disjunctive clauses for any given attribute; that is, multiple views of the same attribute. As in perpendicular filtering, views of an attribute may have different encodings. Each view maintains independent include and exclude selections over all values.

The combination of perpendicular and parallel filtering captures the general case in which any view can be cross-filtered on any combination of views, including zero or more views of its own attribute (including itself). The semantics of such filtering may be prohibitively confusing, however, if brushing is used to form mixed negation disjunctive clauses. In particular, contradictions arise when values are exclude-selected in one view but include-selected in another; conjunction can cause selections to cancel each other out partially or completely. An interesting question is whether such complex filtering states are useful (or at least harmless) when considered as part of the analytic process as a whole, in which it is reasonable to expect hypothesized groupings of people, places, dates, etc. to be multiple, overlapping, and evolving. From this perspective, an analyst's desire to express partially conflicting queries may be sensible, but begs the question of how to balance facility of expression with ease of interpretation.

Especially complicated semantics involving partial cancellation arise in cases in which the sets of negated and non-negated values overlap between views. Such cases can be avoided by restricting parallel filtering to a pair of views of opposing polarity, in which selections are constrained to disjoint sets of attribute values. Figure 10 shows an example of this in which events are filtered on sets of attendees and absentees at events: "Show the types of events that either Alice or David attended and either Barry or Cindy did not."

A second option is *sibling filtering*, in which two or more views of the same attribute filter each other. This approach would provide a way to compose sets flexibly, using the sibling views as a collective working space to compose and modify subsets by brushing [25]. For instance, a protracted parallel query might involve three views of people: one filtered on time, another on location, and a third for picking subsets from the other two for use in further filtering of other attributes' views.

A single view could also be used to define multiple clauses. This might involve variations of selection reuse [6] such as a push/pull selection stack for reversibility of brushing within the view itself, or interactive collection of a heap of selections (disjunctive clauses) that supports read and write by any view of a given attribute. Similarly, remembering sets of clauses across different attributes could involve only a subset of the entire exploratory state of a visualization [8], namely the subset of attributes relevant to particular queries. The interesting possibility is to vary, save, and restore the selection sets of other attributes singly or in combination in order to probe the neighborhood around and causal pathways from exploratory substates that are hypothetically similar over different subsets of attribute values.

The question remains how to represent, interact with, and trigger parallel filters. The N-by-N cross-filtering matrix would be clumsy (and ambiguously labeled) for multiple views of the same dimension. A sandbox metaphor or even a metadata-filtered graph (extending the description in section 2.4) could serve as a primary container for dragging around and directionally linking views in order to build conjunctive visual forms during visual exploration.

3 RELATED WORK

The cross-filtered views model, and by extension the conjunctive visual forms model, target the use of coordinated multiple views in order to attack this problem of dimensional scalability in boolean query visualization. A pioneering approach in this regard is the filter-flow representation of boolean queries [26], in which users construct directed graphs of attribute value lists connected by boolean set operations. Users can drill-down into a full data set in terms of conjunctions or disjunctions of attribute values. The rather minimal visual abstraction of literals (as simple text) and other clauses (as simple scrollable lists) means that it is practical to use filter-flow to rearrange clauses into an overall boolean formula. The directedness of the query operation graph allows drill-down into a list of data records in terms of attribute values, but precludes a richer visual exploration process involving cross-filtering between the sets of attribute values themselves.

Much research on visual boolean querying has focused on visual language approaches, such as interactive Venn diagrams, in which data values are displayed as simple glyphs inside overlapping closed regions, e.g. [9]. The small number of literals typically represented in these techniques is in sharp contrast to the needs of data exploration in visual analysis, which call for visualization approaches that target huge data sets, and hence representation of and interaction with potentially large sets of literals arranged in assorted disjunctions and/or conjunctions. Efforts to create a unified visual abstraction of multidimensional sets include InfoCrystal/MetaCrystal [16] and Sparkler [5], both of which use a radial layout with polygon geometries to segregate and connect sets of values of a modest number of dimensions (typically three to six) in terms of relevance or number of occurrences.

Scientific visualization approaches are also starting to incorporate boolean querying as a way to increase the flexibility of visual exploration for high-dimensional numeric data sets, such as from large-scale simulations [12]. The Feature Definition Language (FDL) enables end-users to specify focus+context visualizations using a high-level declarative language [3]. FDL statements declare how to visually encode features—subsets of data attributes of user interest—as an arbitrary boolean function of object and/or region selections. Users can explore and drill-down into complex multidimensional structures by evolving a collection of features (FDL statements) tailored to data patterns observed along the way. Whereas FDL uses disjunctive normal form to model the visual representation of data as a function of interaction, CVF uses conjunctive normal form to model interaction with data as a function of visual representation. Visual exploration under the CVF model occurs at a high level of abstraction and symmetry, in which lightweight interactions drive not only dynamic querying but
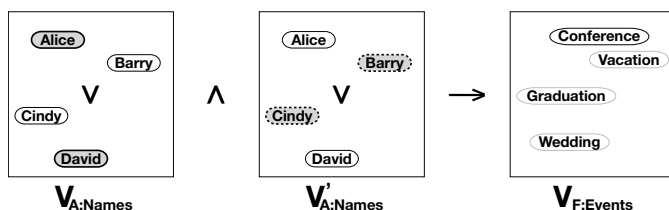

Fig. 10. Parallel filtering. Highlighting of events (in $V_{F:Events}$) depends on a conjunction of two different disjunctions of names (in $V_{A:Names}$, $V'_{A:Names}$). The disjunctions are non-overlapping and of opposite polarity.
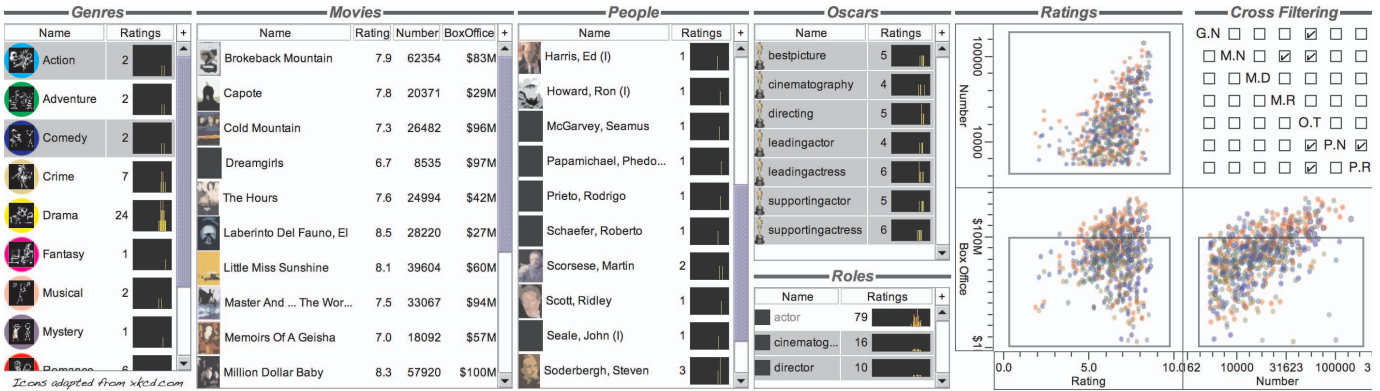
Fig. 11. Cross-filtering in Cinegraph. A sequence of value selections and filter toggles have been used to display two related queries simultaneously: the cinematographers and directors of any award-winning movie, and award-winning movies that earned less than $100M at the box office.

also the rapid specification of logical filtering/highlighting dependencies between views. This makes CVF well-suited for efficient exploration of the large number of discrete idiosyncratic relationships that tend to pervade nominal data at all scales. This is also in contrast with the relatively small number of continuous region patterns typical of quantitative data, which necessitates the greater (but slower) expressiveness of the general boolean formulas used in FDL.

Although boolean queries have long been known to be hard to interpret [15], evaluation (such as that of the Karnaugh map visual tool KMVQL [7]) suggest that visualization may be an effective means to comprehend multidimensional set relationships. The difficulty of composing and remembering sequences of set queries in multidimensional visualization tools—such as the "out-of-sight, out-of-mind" problem observed during evaluation of the hotels visualization [23]— suggests that the usability of boolean query visualization approaches for protracted visual exploration remains a major challenge. The increased expressiveness of conjunctive visual forms likely exacerbates this challenge, but also provides a framework for evaluating promising combinations of interactions and representations.

4 FUTURE WORK: EXTENDING CURRENT TOOLS

Ongoing development of the conjunctive visual forms model is motivated by the successes and shortcomings of current visual analysis tools. An immediate goal of future work will be to implement examples in order to demonstrate and evaluate the full set of interactive manipulations predicted by the model. Here we briefly look at two visual analysis tools that already support a subset of these manipulations, and hence are prime candidates for extension under the model.

Cross-filtered views [22] have been employed in visualizations of a wide variety of multidimensional data sets, including those from the InfoVis 2007 contest and all four VAST 2008 mini-challenges. The Cinegraph visualization [21] supports visual exploration of the small subset of the Internet Movie Database created for the InfoVis 2007 contest. Cross-filtering in Cinegraph can be used to express complex set queries about movies, genres, awards, release dates, ratings, and people with different roles, as shown in figure 11. Selecting literals happens by clicking rows in tables. Selecting disjunctions happens by toggling checkboxes in a cross-filtering matrix. The people view is filtered to show only cinematographers or directors associated with movies that won an award: $((award = bestpicture) \lor \cdots \lor (award = supportingactress)) \land ((role = cinematographer) \lor ((role = director)) \rightarrow V_{People}$. The movie view is filtered to show only movies that won an award but earned less than $100M at the box office: $((award = bestpicture) \lor \cdots \lor (award = supportingactress)) \land (\$0 < boxoffice < \$100M)) \rightarrow V_{Movies}$. The genre and role views are filtered to show only those values associated with a movie that won an award, e.g. $((award = bestpicture) \lor \cdots \lor (award = supportingactress)) \rightarrow V_{Genres}$, and thereby reveal the totals and distribution of genres and roles involved in award-winning movies.

In this example, strictly perpendicular filtering limits queries to cases in which it makes sense for disjunctions of an attribute to be modified (or remain fixed) in the same way for all queries. A single view per attribute precludes parallel filtering. Moreover, each query is limited to a disjunction of include-selected award types, yet there are many useful queries that cannot be expressed easily without exclude-selection: "show movies that won best picture but *not* best director"; "show people who have been in adventures *and* comedies"; "show movies in which two people acted *together* but a third had *no* role". Complementing disjunctions is also highly desirable: "now show awards won in all the *other* genres". We are currently exploring how to represent negation of literals and disjunctions (entire views) in Cinegraph and other cross-filtered visualizations in order to support these kinds of queries. We are also exploring uses of CVF in well-established styles of multidimensional visualization such as parallel coordinate plots and scatter plot matrices (figure 12).

Jigsaw [17] is a domain-specific tool for exploring entity co-occurrences in a collection of report documents. Attributes are displayed in a set of simple lists in which the rows of co-occurring values are cross-linked in the manner of a parallel coordinate plot (figure 13). The analyst can select arbitrary subsets of values in each list, thereby highlighting the values and displaying the corresponding cross-links. Jigsaw also provides interactions to filter values within and across lists, by clicking rows to expand or contract co-occurrences. These additional interactions correspond to combinations of selecting literals, rearranging literals, and reflective filtering. The latter has the effect of visually associating selected and unselected values of each attribute, using an occurrence frequency color scale to fill unselected rows. Jigsaw also allows disjunction rearrangement by adding and removing lists and by switching the attribute shown in each list. The ability to select the same attribute for multiple lists allows parallel filtering (limited by screen width), but not perpendicular filtering. A full and
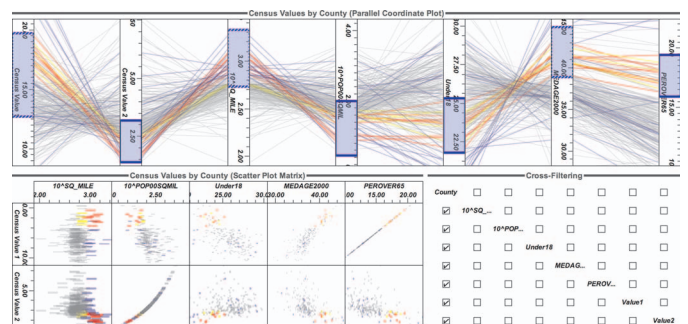

Fig. 12. Cross-filtering seven dimensions of census data using toggled range selections in a parallel coordinate plot and a scatter plot matrix.
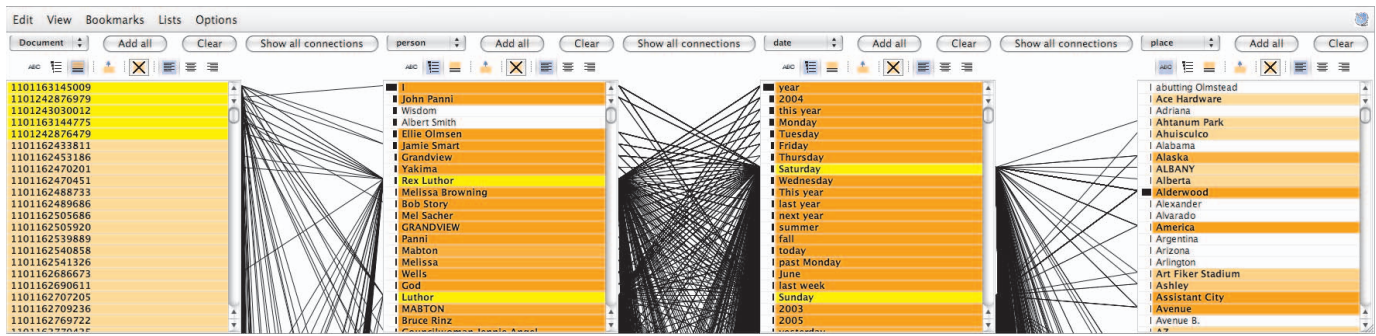
Fig. 13. Visual querying in the list view in Jigsaw, showing documents involving either weekend day and any person with partial name "Luthor".

careful comparative analysis of the multidimensional boolean querying capabilities in Cinegraph and in Jigsaw is a clear next step.

5 CONCLUSION

In conjunctive visual forms, each disjunction represents the collective selection state of a view. Composing a set of disjunctions corresponds to building a multidimensional query as a function of a selected set of values in each of the dimensions. By limiting possible compositions of disjunctions to conjunctive normal form, we are in effect constraining the space of possible coordinated multiple view visualization designs to a highly symmetric subspace. This subspace is nevertheless a superspace of visualization designs possible using cross-filtered views and similar approaches. A balance of concerns for both increased expressiveness of interaction and increased symmetry of visual abstraction gives rise to compelling directions toward understanding fundamental structures and pathways in the synthesis of highly flexible yet widely usable interactive tools for visual exploration.

ACKNOWLEDGMENTS

REFERENCES

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

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

[3] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualization*, pages 239–248, Grenoble, France, 2003. Eurographics Association.

[4] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1141–1148, November/December 2008.

[5] S. Havre, E. Hetzler, K. Perrine, E. Jurrus, and N. Miller. Interactive visualization of multiple query qesults. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 105–112, 2001.

[6] J. M. Heer. *Supporting Asynchronous Collaboration for Interactive Visualization*. PhD thesis, EECS Department, University of California, Berkeley, December 2008.

[7] J. Huo and W. Cowan. Comprehending boolean queries. In *Proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV)*, pages 179–186, Los Angeles, CA, 2008. ACM.

[8] T. J. Jankun-Kelly, M. Kwan-Liu, and M. Gertz. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):357–369, March/April 2007.

[9] S. Jones. Graphical query specification and dynamic result previews for a digital library. In *Proceedings of the Symposium on User Interface Software Technology (UIST)*, pages 143–151, San Francisco, CA, 1998.

[10] 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.

[11] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 271–278, October 1995.

[12] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. *Computer Graphics Forum*, 27(3):775–782, May 2008.

[13] P. Proulx, S. Tandon, A. Bodnar, D. Schroh, R. Harper, and W. Wright. Avian flu case study with nSpace and GeoTime. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 27–34, Baltimore, MD, October/November 2006.

[14] 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.

[15] R. Shepard, C. L. Hovland, and H. M. Jenkins. Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 75:1–42, 1961.

[16] A. Spoerri. Coordinated views and tight coupling to support meta searching. In *Proceedings of Coordinated and Multiple Views (CMV)*, London, UK, July 2004.

[17] 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.

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

[19] F. B. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Many eyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, November/December 2007.

[20] M. O. Ward. Creating and manipulating n-dimensional brushes. In *Proceedings of the Joint Statistical Meeting*, pages 6–14, 1997.

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

[22] C. Weaver. Multidimensional visual analysis using cross-filtered views. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 163–170, Columbus, OH, October 2008.

[23] 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*, 6(1):89–103, February 2007.

[24] G. Wills. Selection: 524,288 ways to say 'this is interesting'. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 54–60, October 1996.

[25] M. A. Wright and J. C. Roberts. Click and brush: A novel way of finding correlations and relationships in visualizations. In *Proceedings of Theory and Practice of Computer Graphics*, pages 179–186. Eurographics, 2005.

[26] D. Young and B. Shneiderman. A graphical filter/flow representation of boolean queries: A implementation and experiment. *Journal of the American Society for Information Science*, 44(6):327–339, July 1993.