DOI:10.1145/2133806.2133821

Article development led by acmqueue
queue.acm.org

A taxonomy of tools that support the fluent and flexible use of visualizations.

BY JEFFREY HEER AND BEN SHNEIDERMAN

Interactive Dynamics for Visual Analysis

THE INCREASING SCALE and availability of digital data provides an extraordinary resource for informing public policy, scientific discovery, business strategy, and even our personal lives. To get the most out of such data, however, users must be able to make sense of it: To pursue questions, uncover patterns of interest, and identify (and potentially correct) errors. In concert with data-management systems and statistical algorithms, analysis requires contextualized human judgments regarding the domain-specific significance of the clusters, trends, and outliers discovered in data.

Visualization provides a powerful means of making sense of data. By mapping data attributes to visual properties such as position, size, shape, and color, visualization designers leverage perceptual skills to help users discern and interpret patterns within data.[4] A single image, however, typically provides answers to, at best, a handful of questions. Instead, visual analysis typically progresses in an iterative process of view creation, exploration, and refinement. Meaningful analysis consists of repeated explorations as users develop insights about significant relationships, domain-specific contextual influences, and causal patterns. Confusing widgets, complex dialog boxes, hidden operations, incomprehensible displays, or slow response times can limit the range and depth of topics considered and may curtail thorough deliberation and introduce errors. To be most effective, visual analytics tools must support the fluent and flexible use of visualizations at rates resonant with the pace of human thought.

The goal of this article is to assist designers, researchers, professional analysts, procurement officers, educators, and students in evaluating and creating visual analysis tools.

We present a taxonomy of interactive dynamics that contribute to successful analytic dialogues. The taxonomy consists of 12 task types grouped into three high-level categories, as shown in the accompanying table: data and view specification (visualize, filter, sort, and derive); view manipulation (select, navigate, coordinate, and organize); and analysis process and provenance (record, annotate, share, and guide). These categories incorporate the critical tasks that enable iterative visual analysis, including visualization creation, interactive querying, multiview coordination, history, and collaboration. Validating and evolving this taxonomy is a community project that proceeds through feedback, critique, and refinement.

Taxonomy of interactive dynamics for visual analysis.

Data and View Specification	**Visualize** data by choosing visual encodings.
	Filter out data to focus on relevant items.
	Sort items to expose patterns.
	Derive values or models from source data.
View Manipulation	**Select** items to highlight, filter, or manipulate them.
	Navigate to examine high-level patterns and low-level detail.
	Coordinate views for linked, multidimensional exploration.
	Organize multiple windows and workspaces.
Process and Provenance	**Record** analysis histories for revisitation, review, and sharing.
	Annotate patterns to document findings.
	Share views and annotations to enable collaboration.
	Guide users through analysis tasks or stories.

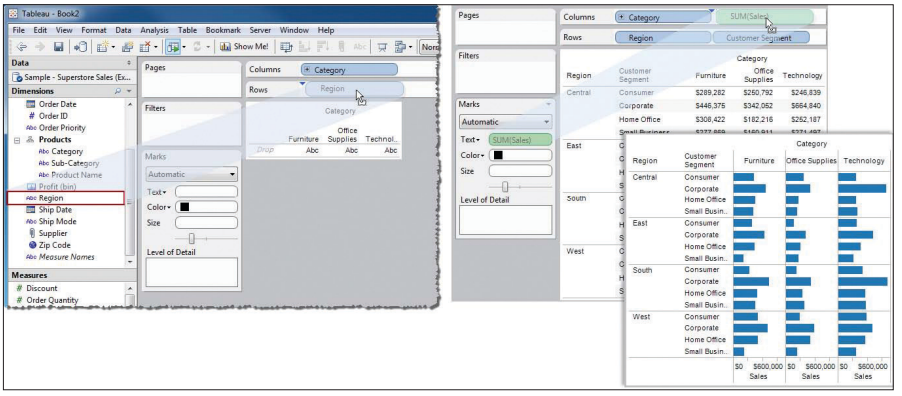


Figure 1. Visual encoding via drag-and-drop actions in Tableau.

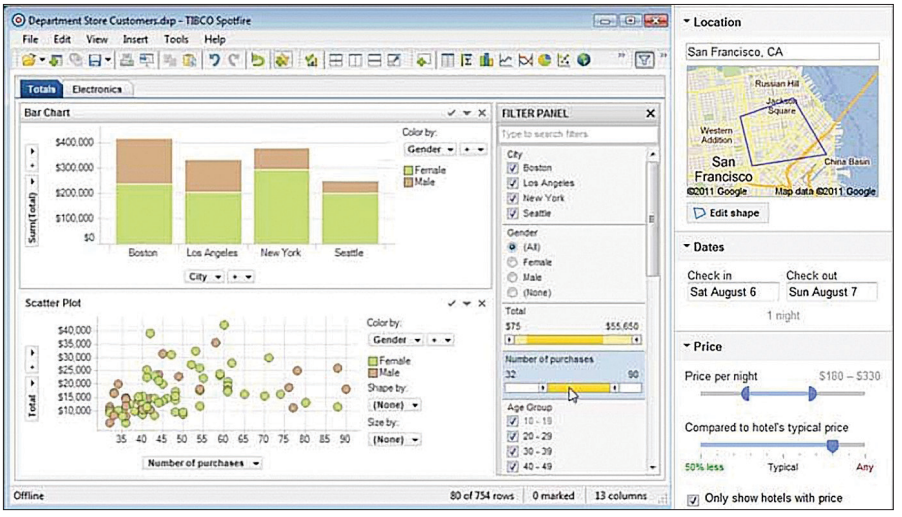


Figure 2. Examples of dynamic query filter widgets from Spotfire (left) and Google Hotel Search (right).

Our focus on interactive elements presumes a basic familiarity with visualization design. The merits and frailties of bar charts, scatter plots, timelines, and node-link diagrams, and of the visual encoding decisions that underlie such graphics, are certainly a central concern, but we will largely pass over them here. A number of articles and books address these topics in great detail,[4,5,20] and we recommend them to interested readers.

Within each branch of the taxonomy, we describe example systems that exhibit useful interaction techniques. To be clear, these examples do not constitute an exhaustive survey; rather, each is intended to convey the nature and diversity of interactive operations. Throughout the article the term *analyst* refers to someone who uses visual analysis tools and not to a specific person or role. Our notion of analyst encompasses anyone seeking to understand data: traditional analysts investigating financial markets or terrorist networks, scientists uncovering new insights about their data, journalists piecing together a story, and people tracking various facets of their lives, including blood pressure, money spent, electricity used, or miles traveled.

Data and View Specification

To enable analysts to explore large datasets involving varied data types (for example, multivariate, geospatial, textual, temporal, networked), flexible visual analysis tools must provide appropriate controls for specifying the data and views of interest. These controls enable analysts to selectively *visualize* the data, to *filter* out unrelated information to focus on relevant items, and to *sort* information to expose patterns. Analysts also need to *derive* new data from the input data, such as normalized values, statistical summaries, and aggregates.

Visualize. Perhaps the most fundamental operation in visual analysis is to specify a visualization of data: analysts must indicate which data is to be shown and how it should be depicted. Within user interfaces, such visualization "widgets" are often presented in a *chart typology*, a palette of available vi-

sualization templates (bar charts, scatter plots, map views.) into which analysts can slot their data. This method of interaction will be immediately familiar to users of spreadsheet programs: users select a chart type and assign data variables to visual aspects such as the X/Y axes and the size or color of visualized marks. A chart typology has the benefits of simplicity and familiarity, but it also limits the types of possible visualizations.

Visualization system designers have explored alternative approaches. Classic scientific visualization systems[1] use *data-flow graphs*, in which the visualization process is deconstructed into a set of finer-grained operators for data import, transformation, layout, or coloring. However, novel designs often require programming expertise to develop new operators for the system. Other systems are based on *formal grammars* that succinctly describe how data should be mapped to visual features. This approach is used by a number of popular data-visualization frameworks such as Leland Wilkinson's *Grammar of Graphics*,[24] ggplot2 for the R statistical analysis platform, and Protovis for HTML5. Formal grammars can be augmented with automated design facilities: a system can generate multiple visualization suggestions from a partial specification. Tableau (née Polaris[19]) enables visualization specification by drag-and-drop operations: analysts place data variables on "shelves" corresponding to visual encodings such as spatial position, size, shape, and color (see Figure 1). This specification is then translated into an underlying formal grammar that determines both the visualization design and corresponding queries to a database.

Fortunately, these methods are not mutually exclusive. Analysts can apply a data-flow system or formal grammar to define new components to include within a chart typology, leveraging the improved expressiveness of the former and the ease of use of the latter. Novel interfaces for visualization specification are still needed, as new tools requiring little to no programming might place custom visualization design in the hands of broader audiences.

Filter. Filtering of data values is intrinsic to the visualization process, as analysts rarely visualize the entirety of a data set at once. Instead, they construct a variety of visualizations for selected data dimensions. Given an overview of selected dimensions, analysts then often want to shift their focus among different data subsets—for example, to examine different time slices or isolate specific categories of values.

Designers have devised a variety of interaction techniques to limit the number of items in a display. Analysts might directly select (for example, "lasso") items in a display and then highlight or exclude them; we discuss these forms of direct view manipulation later. Another option is to use a suite of auxiliary controls, or *dynamic query widgets*,[17] for controlling item visibility (see Figures 2 and 3). The choice of an appropriate widget is largely determined by the underlying data type. Categorical or ordinal data can be filtered using simple radio buttons or checkboxes (when the number of distinct items is small), or scrollable lists, hierarchies, and search boxes with autocomplete (when the number of distinct items is large or contains arbitrary text). Ordinal, quantitative, and temporal data can also be filtered using a standard slider (for a single threshold value) or a range slider (for specifying multiple endpoints). When coupled with real-time updates to the visualization, these widgets allow rapid and reversible exploration of data subsets. In Figure 2, Spotfire

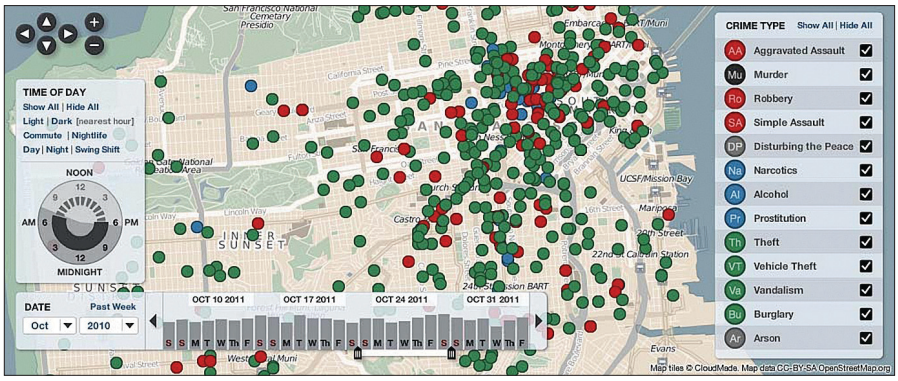


Figure 3. Zoomable map from CrimeSpotting.org.

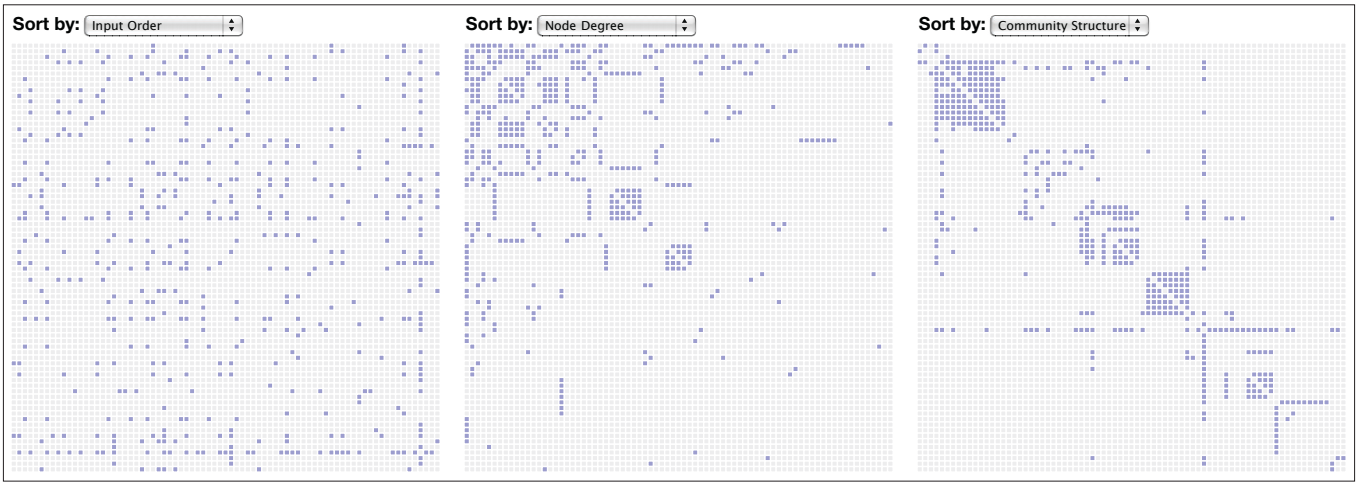
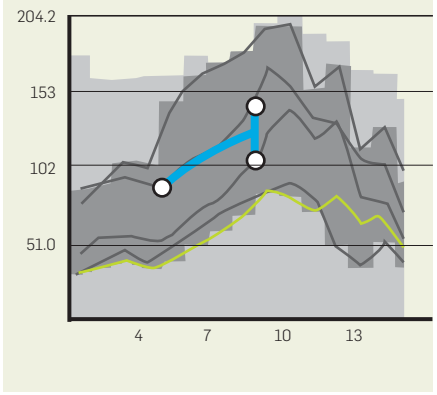


Figure 4. Reorderable matrices.

(left) provides a variety of controls for filtering visualized data: checkboxes and radio buttons filter categorical variables, while range sliders filter numerical values; on the right, Google Hotel Search provides widgets for geographic, date, and price ranges. Query controls can be further augmented with visualizations of their own: Figure 3 includes a range slider for dates augmented with a histogram of underlying values.

Expert analysts also benefit from more advanced functionality. For example, a search box might support sophisticated query mechanisms, ranging in complexity from simple keyword search to a full-fledged query language. Filtering also interacts with other operations: filtering widgets may operate over data sorted in a user-specified manner (see the next section), or users might create derived values (as we will discuss) and filter based on the results.

Figure 5. Querying time-series by slope in TimeSearcher.[12]



Sort. Ordering (or sorting) is another fundamental operation within a visualization. A proper ordering can effectively surface trends and clusters of values or organize the data according to a familiar unit of analysis (days of the week, financial quarters, and so on). The most common method of ordering is to sort records according to the value of one or more variables. Ordering becomes more complicated in the case of multiple view displays, in which both entire plots and the values they contain may be sorted to reveal patterns or anomalies. Sorting values consistently across plots (for example, by their marginal mean or median values) can reveal patterns while facilitating comparison among plots.

Some data types (for example, multivariate tables, networks) do not lend themselves to simple sorting by value. Such data may require more sophisticated *seriation* methods[24] that minimize a distance measure among items. The goal is to reveal underlying structure within the data. Figure 4 shows a matrix-based visualization of a social network. On the left, a matrix plot of a social network conveys little structure when the rows and columns (representing people) are sorted alphabetically. Interactively reordering the matrix by node degree reveals more structure (center). Permuting the matrix by network connectivity reveals underlying clusters of communities (right).

Derive. As an analysis proceeds in iterative cycles, users may find that the input data is insufficient: variables may need to be transformed or new attributes derived from existing values. Common cases include normalization or log transforms to enable more effective value comparisons. Derived measures are often used to summarize the input data, ranging from descriptive statistics (mean, median, variance) to model fitting (regression curves) and data transformation (group-by aggregation such as counts or summations). Often this functionality is provided via a *calculation language*, similar to those found in spreadsheets or database query languages.

Improved derivation methods present a promising frontier for visual analytics research. How can visual tools support flexible construction of more advanced models or derived values? Analysts might define patterns using programming-by-demonstration

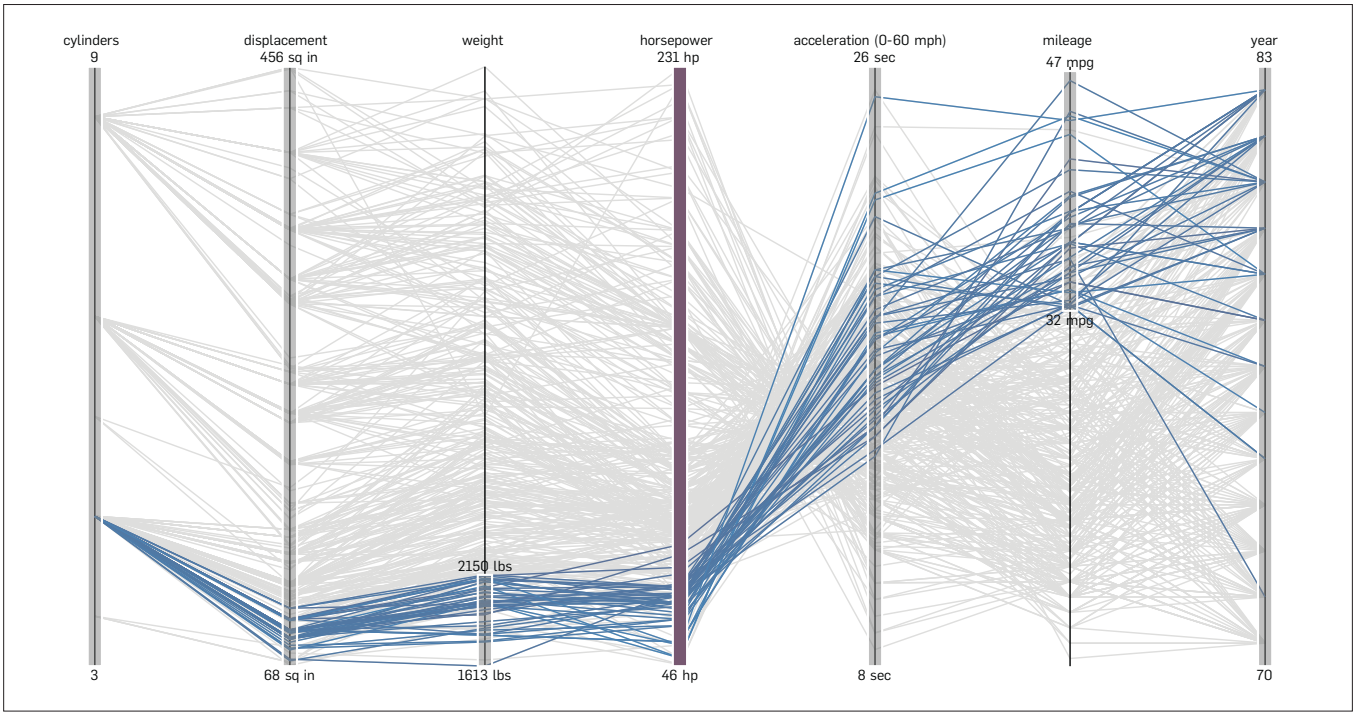


Figure 6. Selection queries in parallel coordinates.

methods. Or, visual tools might automatically fit applicable statistical models to the data based on the current visualization state. For example, the nesting of variables within common "pivot" displays could be mapped to the structure of a linear model. More principled frameworks that wed visualization to modeling and forecasting are still emerging.

View Manipulation

Once analysts have created a visualization, they should be able to manipulate the view to highlight patterns, investigate hypotheses, and drill down for more details. Analysts must be able to *select* items or data regions to highlight, filter, or operate on them. Large information spaces may require analysts to scroll, pan, zoom, and otherwise *navigate* the view to examine both high-level patterns and fine-grained details. Multiple, linked visualizations often provide clearer insights into multidimensional data than do isolated views. Analysis tools must be able to *coordinate* selections across multiple views and *organize* the resulting dashboards and work spaces.

Select. Pointing to an item or region of interest is common in everyday communication because it indicates the subject of conversation and action. In the physical world, people coordinate their gestures, gaze, and speech to indicate salient items. For example, different hand gestures can communicate angle (oriented flat hand), height (horizontal flat hand), intervals (thumb and index finger in "C" shape), groupings (circling a region), and forces (accelerating fist).[11] In visual analysis, reference (or *selection*) remains of critical importance, but is realized through a more limited set of actions, such as clicking or lassoing items of interest.

Common forms of selection within visualizations include mouse hover, mouse click, region selections (for example, rectangular and elliptical regions, or free-form "lassos"), and area cursors (for example, "brushes"[2] or dynamic selectors such as the bubble cursor,[7] which selects the item currently closest to the mouse pointer).

Selections can vary in terms of their expressive power. Most interfaces support selections of a collection of items. Though this approach is easy to implement, it does not allow analysts to specify higher-level criteria. A more powerful, albeit more complex, approach is to support selections as queries over the data. Maintaining query structure increases the expressiveness of visualization applications. For example, drawing a rectangle in a chart may specify a range query over the data variables represented by the x and y axes. The resulting selection criteria can then be saved and applied to dynamic data (updating items may enter or exit a query region) or to a completely different visualization. Examples include querying stock-price changes in TimeSearcher[12] (see Figure 5) and attribute ranges in parallel coordinates displays[13] (Figure 6). In Figure 5 an angular selection tool specifies a target slope (rate of change) and tolerance for a collection of stock prices. All time series with a similar slope over the queried time range are selected; shaded regions show envelopes of minimum and maximum values. The widget operates directly on the visualization: dragging the widget from left to right interactively queries other time windows. In Figure 6 parallel coordinates plot multidimensional data as line segments among parallel axes. Here, an analyst has dragged along the axes to create interactive selections that highlight automobiles with low weight and high mileage.

Designing more expressive selection methods remains an active area of research. Enhanced selections might incorporate data semantics (for example, values, hierarchy, or clusters) to enable guides or "snap-to" actions. Nuanced selections might be specified with more fluid gestures. Of course, selection need not be limited to the mouse and keyboard: input modalities such as touch and speech might enable new, effective forms of selection.

Navigate. How analysts navigate a visualization is in part determined by where they start. One common pattern of navigation adheres to the widely cited visual information-seeking mantra: "Overview first, zoom and filter, then details-on-demand."[18] Analysts may begin by taking a broad view of the data, including assessment of prominent clusters, outliers, and potential data-quality issues. These orienting actions can then be followed by more specific, detailed investigations of data subsets. A common example is geographic maps. The map in Figure 3 depicts criminal activity by time and region. It shows all crimes committed after dark during the last week of October 2011. Dynamic query widgets enable filtering by time of day (left), date span (bottom), and type of crime (right). Pan (drag) and zoom (buttons and scroll wheel) controls enable view navigation. As an analyst zooms in on the map, the circular crime markers gain detailed labels—a form of *semantic zooming*.

Of course, starting with an expansive overview is not always advisable. A legal analyst researching for an upcoming trial may be wise to forego an overview of the entire history of U.S. court decisions. Instead, the analyst might start with the legal decisions most relevant to the current case, perhaps determined by keyword search, and expand the investigation to other, cited decisions. This form of navigation can be summarized as "Search, show context, expand on demand."[21]

In either case, visualizations often function as manipulable *viewports* onto an information space. Common examples include scrolling or panning a display via scrollbars or mouse drag, and zooming among different levels using a zoom slider or scroll wheel (Figure 3). Zooming need not follow a strict geometric metaphor: semantic zooming[3] methods can modify both the amount of information shown and how it is displayed as analysts move among levels of detail.

To aid navigation further, researchers have developed a variety of *focus plus context* methods. These displays provide a detailed view of a high-interest data region while retaining surrounding context to help keep analysts oriented. A second key idea is the use of *overview and detail* displays. For example, a geographic visualization might include a large zoomed-in map (the detail), while a smaller, zoomed-out map includes a rectangle showing the position of the zoomed-in view within the broader terrain (the overview). In this case, the detail view provides the focus, and the overview provides context.

A different approach is to use *distor-*

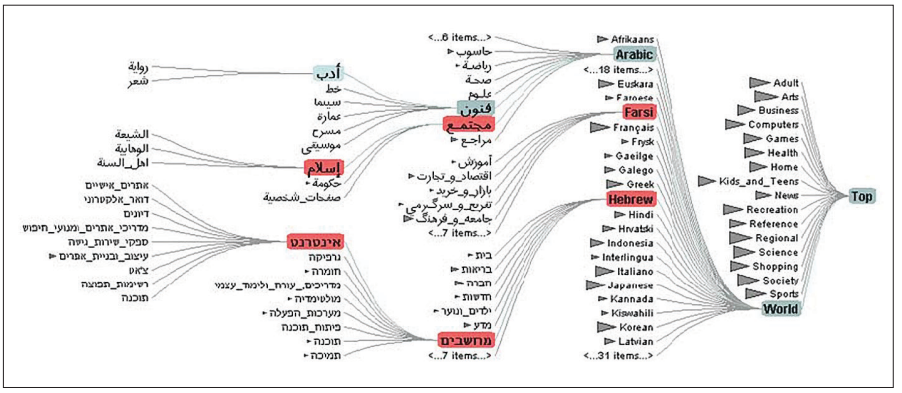


Figure 7. Degree-of-interest tree of a taxonomy with 600k items.[8]

tion techniques to demagnify contextual regions. A simple example is the Mac OS X dock, which uses 1D fisheye distortion to show common applications; more sophisticated methods employ distortion in multiple dimensions. While often visually intriguing, complex distortion methods have yet to prove their worth in real-world applications: viewers can become disoriented by nonlinear distortions, which show no significant performance improvement over simpler methods such as zooming.

In addition to manipulating display space, focus-plus-context methods can be applied directly to the data. The goal is to identify which data items are currently of high interest (focus), which are of high importance regardless of the current focus (context), and which can be safely removed from view. *DOI* (*degree-of-interest*) functions[8,21] calculate scores for information content based both on general importance (for example, top-level categories in a hierarchy or high-centrality nodes in a graph) and current interest (for example, as indicated by mouse clicks, search queries, or proximity to other high-interest items). The distribution of DOI scores can then be used to control the visibility of items based on the current view size and context of interaction, as in Figure 7. As analysts click on or search for different items, the DOI scores dynamically update to reveal relevant unseen data or hide irrelevant detail. A model of the analyst's current interest filters the display to the most relevant items. Low-interest items are elided but still accessible through aggregate representations. The interest estimates update as an analyst explores the taxonomy, initiating animated transitions between different views of the data.

Visualizations can also provide cues to assist analysts' decisions of where and how to navigate. An important challenge is to show selected items, even when they are not in view. For example, the results of a text search that are not currently in view might be shown by markers in the scrollbar or the periphery of the display.

Coordinate. Many analysis problems require *coordinated multiple* views that enable analysts to see their data from different perspectives. A public policy analyst studying educational attainment might produce a bar chart of people's ages, a map of locations, a textual list with education history, and a scatter plot showing income vs. education. By selecting a single item or a group in one view, analysts might see related details or highlighted items in the other views.

Multiview displays can facilitate comparison. For example, Edward Tufte[20] advocates the use of *small multiples*: a collection of visualizations placed in spatial proximity and typically using the same measures and scales. Small multiples enable rapid comparison of different data dimensions or time slices.

Alternatively, multiple view displays can use a variety of visualization types—such as histograms, scatter plots, maps, or network diagrams—to show different projections of a multidimensional data set. An analyst constructs a complex patchwork of interlinked tables, plots, and maps in Figure 8 to analyze the outcomes of elections in Michigan.[23] Annotations indicate how selected data items correspond between visualization views. Accompanying items such as legends, histogram sliders, and scrollbars with highlighting markers also provide views onto the data.

Multiview displays also enable multidimensional exploration. *Brushing and linking* is the process of selecting (brushing) items in one display to highlight (or hide) corresponding data in the other views.[2] In Figure 9, a baseball analyst makes selections in one plot and corresponding items highlight in the others. On the left, selecting high-income players (top-right plot) shows little dependence on career length or fielding ability, but correlates with hitting performance. On the right, selecting the cluster of players who make more assists than putouts (middle-left plot) reveals a strong dependence on position. Each visualization can thus serve as an input channel for revealing patterns across a data set. By allowing analysts to assess how patterns in one view project onto the others, linked selection enables multidimensional reasoning. Analysts may wish to coordinate views in variety of ways: selecting items in one view might highlight matching records in other views, or instead provide filtering criteria to remove information from the other displays. Linked navigation provides an additional form of coordination: scrolling or zooming one view can simultaneously manipulate other views.

Though comparing multiple visualizations requires viewers to orchestrate their attention and mentally integrate patterns among views, this process is often more effective than cluttering a single visualization with too many dimensions. Future studies of how analysts construct multiview displays and specify coordination behaviors (for example, highlighting, filtering) could provide designers with an understanding of how to build more effective tools. In addition, if designers ensure that rich multiview displays stay understandable, analysts are more likely to make compelling insights. Newcomers to an analysis, or even seasoned analysts simply returning from a coffee break, may become confused by the number of views and the potentially complicated set of coordinated queries between them. Vi-

sual analytics systems that provide access to coordination settings and replay the history of view construction can enhance understanding.

Organize. When analysts make use of multiple views they must manage a collection of visualizations. As in traditional window-based interfaces, analysts may wish to open, close, maximize, and lay out different components. As manual manipulation can be tedious, well-designed visual analytics tools simplify the organization of visualization views, legends, and controls. A tiled layout approach allows analysts with sufficiently large displays to see all the information and selectors at once, minimizing distracting scrolling or window operations. The coordination across windows means that slider movements or checkbox selections will cause all views to update, allowing rapid exploration.

Typical systems allow analysts to add views in ways that make modest changes to the existing window organization. An alternative approach is to add a new tab that contains a new plot, so analysts can switch between the first and second set of windows. A common feature is to add trellised views, so multiple visualizations can be created at once—for example, separate bar charts showing age distributions in different regions.

More advanced systems might aid this process through automated support that enables multiple windows to be opened/closed as a group and lays them out in orderly ways. Useful methods include standard scatter-plot matrices (showing all pairs of scatter plots) or custom generation of related views of interest (for example, of data variables correlated to the visualized attributes). Desirable features are automatic (re)sizing as views are added or removed and layout routines to place related views in spatial proximity.

As larger and multiple displays become more common, layout organization tools will become decisive factors in creating effective user experiences. Similarly, the demand for tablet and smartphone visualizations will promote innovation in layout organizations that are compact and reconfigurable by simple gestures. Zooming, panning, flipping, and sequencing strategies will also improve analyst experiences and facilitate effective presentations to others.

Process and Provenance

Visual analytics is not limited to the generation and manipulation of visualizations—it involves a process of iterative data exploration and interpretation. As a result, visual analytics tools that provide facilities for scaffolding the analysis process will be more widely adopted. Tools should preserve analytic provenance by keeping a *record* of analyst actions and insights so the history of work can be reviewed and refined. If analysts can *annotate* patterns, outliers, and views of interest, they can document their observations, questions, and hypotheses. Analysts should be empowered to *share* results and discuss with colleagues, coordinate the work of multiple groups, or support processes that may take weeks and months. Moreover, analysis tools can explicitly *guide* novices through common analysis tasks, provide progress indicators, or lead viewers through an analysis story.

Record. When analyzing data with visualizations, users regularly traverse the space of views in an iterative fashion. Exploratory analysis may result in a number of hypotheses, leading to multiple rounds of questions and answers. To support iterative analysis, visual analysis tools can record and visualize analysts' *interaction histories*. At a minimum, applications should provide basic undo and redo support. By modeling the space of user actions (view specifications, sorting, filtering, or zooming), rich logs can be constructed and visualized.[6,9] Common visual representations of analytic actions include both chronological ("timeline") and sequential ("comic strip") views. As shown in Figure 10, a "comic strip" display retraces the

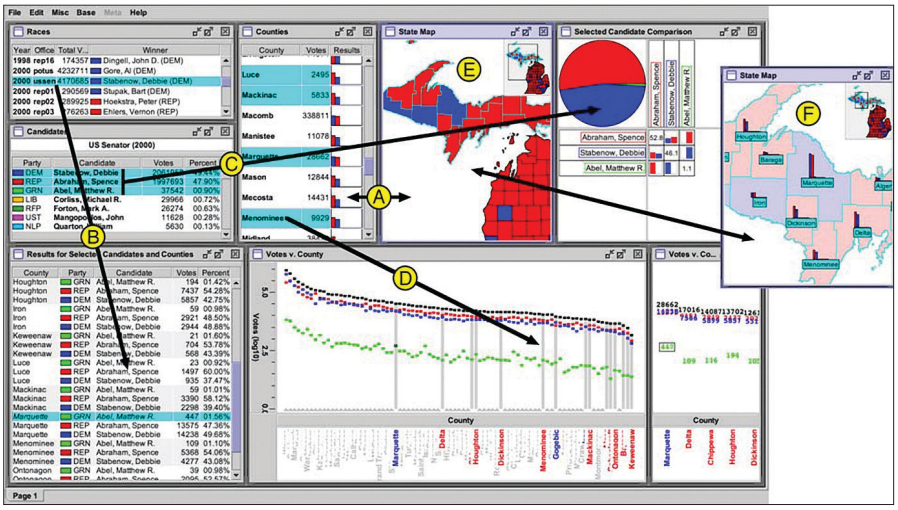


Figure 8. Multiple coordinated views in Improvise.[23]

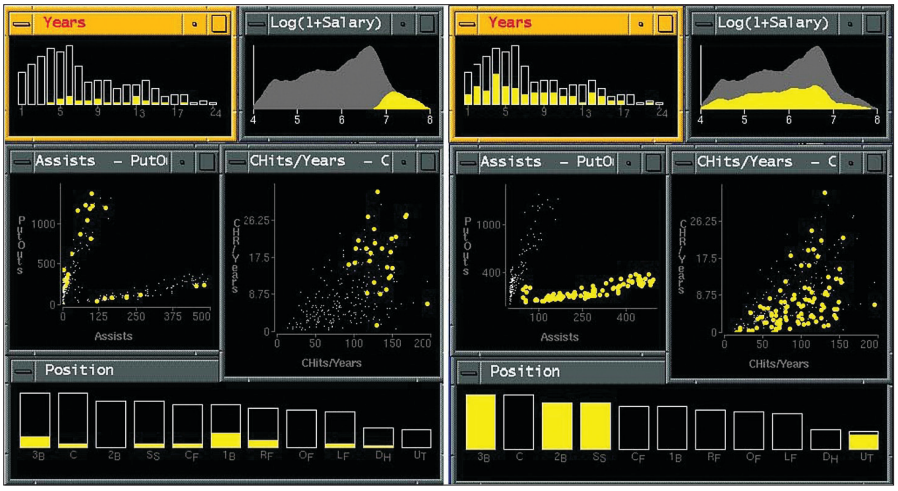


Figure 9. Brushing and linking of baseball statistics in GGobi.

Figure 10. Visual analysis history.[9]

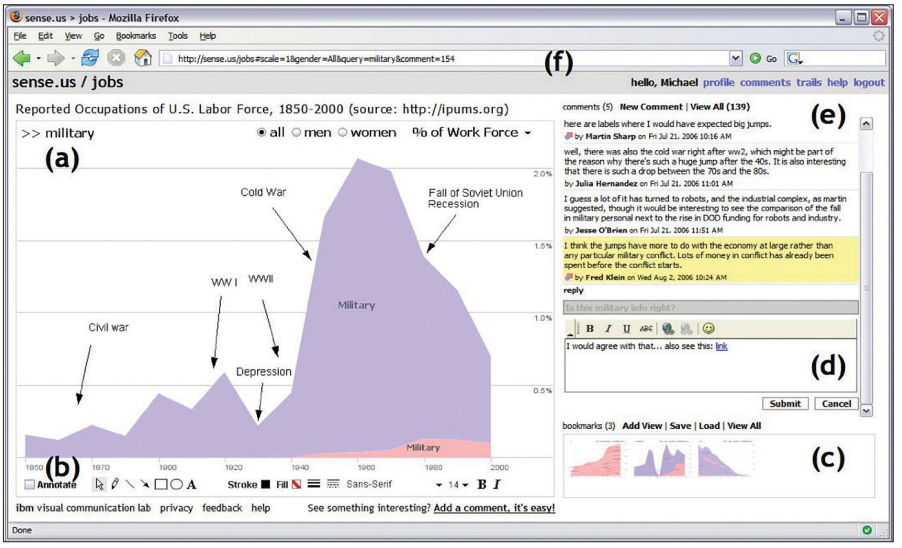


Figure 11. Collaborative visual analysis in Sense.us.[10]

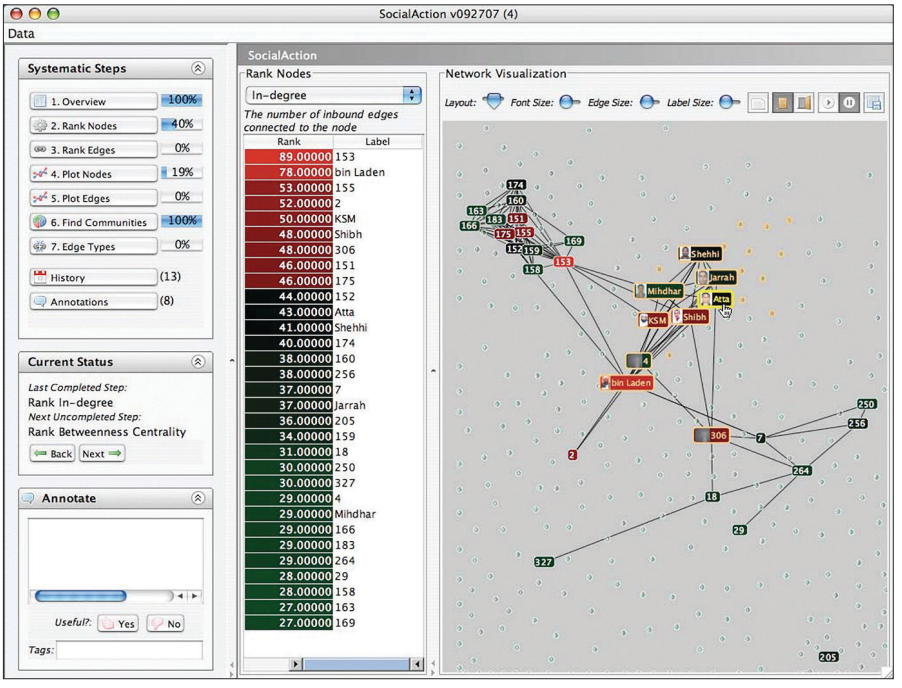


Figure 12. Systematic yet flexible analysis in SocialAction.[15]

steps taken in a visual analysis of business operations data.

Visual histories can support a range of interactions. First, histories provide a convenient mechanism to revisit prior analysis states and resume incomplete explorations. Adding metadata such as comments, tags, or ratings to states can facilitate later review and sharing. Interactive histories can also capture a repeatable sequence of operations that can be named and saved as a reusable macro. This feature enables analysts who are dealing with many similar data sets to automate their efforts. Histories might spur sharing: analysts can export selected analysis trails, ranging from screen shots to interactive presentations, to external media. Finally, histories also provide a means to study analysts and model analytic processes.

Annotate. Interactive visualizations often serve not only as data-exploration tools, but also as a means for recording, organizing, and communicating insights gained during exploration. One option is to allow textual annotation of states within a visual history. More expressive annotations are possible through direct interaction with the view, using the selection techniques discussed earlier. Analysts may wish to "point" to specific items or regions within a visualization and associate these annotations with explanatory text or links to other views.

Freeform *graphical annotations* provide one expressive form of pointing.[10] Drawing a circle around a cluster of items or pointing an arrow at a peak in a graph can direct the attention of viewers. The angle or color of the arrow or shape of the hand-drawn circle may communicate emotional cues or add emphasis. Although such drawings allow a high degree of expression, they lack an explicit tie to the underlying data. Free-form annotations implemented as vector graphics can persist over geometric transformations such as panning and zooming, but if they are not "*data-aware*," then they may become meaningless in the face of operations such as filtering or aggregation.

Annotations can be made data-aware when realized as selections. These selections can be represented as a set of selected items, a declarative query, or both. Data-aware annotations allow a pointing intention to be reapplied to different views of the same data, enabling reuse of refer-

ences across different choices of visual encodings. As data-aware annotations are machine readable, they might also be used to enable search or identify data subsets of high interest.

Share. Researchers in visual analytics often focus on the perceptual and cognitive processes of a single analyst. In practice, real-world analysis is also a social process that may involve multiple interpretations, discussion, and dissemination of results.[10,22] To support the analysis life cycle, visual analytics tools should support social interaction. At minimum, tools must be able to export views or data subsets for sharing and revisitation. Figure 11 shows sense.us,[10] one example of a collaborative visual analysis tool incorporating view sharing, annotation, and discussion. The system consists of (a) an interactive visualization, (b) a set of graphical annotation tools, (c) bookmark trails for saved views, (d) text-entry field for adding comments (bookmarks can be dragged onto the text field to link views to a comment), (e) textual comments attached to the current view, and (f) a shareable URL that is updated automatically as the visualization state changes.

A simple but effective aid to collaboration is view sharing via *application bookmarking*: a visual analytics system should be able to model and export its internal state. Unlike a static screen shot, bookmarking enables analysts to take up an exploration where their collaborators left off. View sharing often takes the form of a URL or similar identifier that allows a collaborator to navigate quickly to a view of interest. Seeing an identical view provides collaborators with a common ground for discussion. Annotation methods can be applied within such views to further collaboration. One challenge for effective view sharing concerns dynamic data: should a bookmarked view maintain a snapshot to historical data, provide access to the most current data, or both?

Another method of sharing and dissemination is to *publish* a visualization. Commercial tools such as Spotfire and Tableau can publish visualization dashboards as interactive Web pages with support for selection, search, and drill-down to enable some amount of follow-up analysis. Services such as IBM's Many Eyes[22] can be used to embed visualization applets in external Web sites. While publishing is a necessary condition for broad sharing, it may not be sufficient by itself for engaging viewers. Visualizations embedded within a blog or discussion forum can reach an established audience and may foster discussion more effectively than a centralized site.

Other collaborative concerns depend on the context of use. Are collaborators working *synchronously* (same time) or *asynchronously* (different time)? Are they *co-located* (same place) or *distributed* (different place)? Each of these configurations may require specialized strategies for access control, presence indicators, and activity awareness.[10,14]

Guide. The exploration process is well understood for some traditional domains. For example, a very simple workflow might remove incomplete data items, sort, select high-value items, and report on these selections. Analysts, however, may need to develop new strategies that are formalized to guide newcomers and provide progress indicators to experts. Visual analysis systems can incorporate *guided analytics* to lead analysts through workflows for common tasks.

Some processes are clearly linear, but many visual analytics tasks require richer *systematic yet flexible* processes that allow analysts to take excursions while keeping track of what they have done. For example, SocialAction[15] organizes social-network analysis into a sequence of activities (for example, rank nodes, plot nodes, find communities); the system allows analysts to skip steps selectively and keeps a record of which steps have been completed. In Figure 12, the panel on the left suggests common steps to structure social network analysis and provides progress indicators. In a related vein, experts often develop visualizations that are used by less knowledgeable team members, in much the same way that spreadsheet macros enable specialists to encode accounting or business practices for others. More research is needed to identify effective visual analytics processes and enable expert analysts to create reusable workflows.

In recent years, journalists have been experimenting with different forms of *narrative visualization*[16] by structuring interactive graphics to tell stories with data. Visualizations from *The New York Times*, *Washington Post*, *The Guardian*, and other news sources often lead the viewer through a linear narrative, guided by supporting text and annotations. In Figure 13, for example, an interactive graphic guides the reader through decades of budget predictions. At a story's conclusion, such visualizations provide interactive controls for further exploration. These narrative structures both communicate key observations from the data and cleverly provide a *tacit tutorial* of the available interactions by animating each component along with the story. By the time the presentation opens up for freeform exploration,

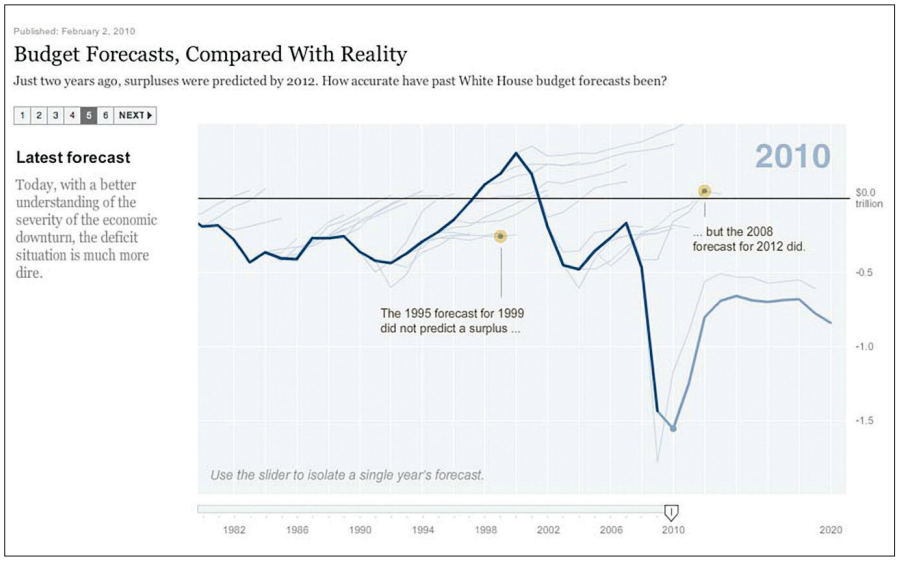


Figure 13. Data storytelling by *The New York Times*.

the viewers have already seen demonstrations of the interactive controls. These forms of narrative visualization demonstrate how guided analytics can help disseminate data-driven stories to a general audience.

Conclusion

We hope this taxonomy and discussion will help advance visual analytics on multiple fronts. For students and newcomers to the field, the taxonomy provides an orientation to the interactive concerns at the heart of visual analysis. We encourage interested readers to consult the systems, books, and papers referenced in this article to develop a deeper understanding. For developers, the taxonomy provides a checklist of items to consider when creating new analysis tools. For researchers, the taxonomy helps highlight critical areas that would benefit from further investigation, including new methods for interactive view specification, a closer integration of visualization and statistical algorithms, and effective approaches to guided analytics.

Of course, by attempting to provide an abstracted picture of a domain, taxonomies may be incomplete. In some cases, we separately categorize aspects that are closely related. Dynamic query widgets enabling data specification often serve as a means of view navigation. Selection techniques are also central to effective annotation schemes.

In other instances, we selectively omit material. For example, we do not go into great depth regarding implementation details. Especially for large datasets, supporting real-time interactivity requires careful attention to system design and poses important research challenges ranging from low-latency architectures to intelligent sampling and aggregation methods. How to best incorporate statistical methods into a visualization environment remains a central challenge; our discussion of derived data only scratches the surface. Other concerns include formatting, cleaning, and integrating data. Incorrect or improperly structured data diverts the energy of trained analysts and presents a significant barrier to newcomers.

These concerns represent active areas of research, and we expect our characterization of the field to evolve in the years to come. We invite the insights and commentary of the visualization, statistics, database, and HCI communities, and eagerly anticipate the continued flowering of improved tools for making sense of the wealth of data that surrounds us.

Acknowledgments

We thank our colleagues and students for providing valuable comments on drafts: Maneesh Agrawala, Jason Chuang, Cody Dunne, John Guerra-Gomez, Pat Hanrahan, Sean Kandel, Diana MacLean, and Kostas Pantazos.

This work was partially supported by National Science Foundation grants IIS-0968521, IIS-1017745, CCF-0964173 and SBE-0915645, NIH-National Cancer Institute grant RC1-CA147489, and ONC-SHARP grant on Cognitive Information Design and Visualization.

Related articles on queue.acm.org

A Conversation with Jeff Heer, Martin Wattenberg and Fernanda Viégas
http://queue.acm.org/detail.cfm?id=1744741

A Tour through the Visualization Zoo
Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky
http://queue.acm.org/detail.cfm?id=1805128

A Conversation with Ed Catmull
http://queue.acm.org/detail.cfm?id=1883592

References

For a complete set of references, see http://queue.acm.org/detail.cfm?id=2146416/

1. Abram, G. and Treinish, L. An extended data-flow architecture for data analysis and visualization. In *Proceedings of the IEEE Conference on Visualization* (1995), 263–270.
2. Becker, R.A. and Cleveland, W. S. Brushing scatterplots. *Technometrics 29*, 2 (1987), 127–142.
3. Bederson, B.B. and Hollan, J.D. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (1994), 17–26; http://doi.acm.org/10.1145/192426.192435.
4. Card, S.K., Mackinlay, J. and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
5. Cleveland, W.S. *The Elements of Graphing Data*. Hobart Press, Lafayette, IN, 1994.
6. Derthick, M. and Roth, S.F. Enhancing data exploration with a branching history of user operations. *Knowledge Based Systems 14*, 1-2 (2001): 65–74.
7. Grossman, T. and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2005), 281-290; http://doi.acm.org/10.1145/1054972.1055012.
8. Heer, J. and Card, S.K. DOITrees revisited: scalable, space-constrained visualization of hierarchical data. *Proceedings of Advanced Visual Interfaces:* (2004), 421–424; http://doi.acm.org/10.1145/989863.989941.
9. Heer, J., Mackinlay, J., Stolte, C. and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008): 1189–1196; http://portal.acm.org/citation.cfm?id=1477066.1477414.
10. Heer, J., Viégas, F. B. and Wattenberg, M. Voyager and voyeurs: supporting asynchronous collaborative information visualization. *Commun. ACM 52*, 1 (Jan. 2009): 87–97; http://doi.acm.org/10.1145/1435417.1435439.
11. Hill, W.C. and Hollan, J.D. Deixis and the future of visualization excellence. In *Proceedings of the IEEE Conference on Visualization*: (1991), 314–320; http://portal.acm.org/citation.cfm?id=949607.949659.
12. Hochheiser, H. and Shneiderman, B. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Info. Visualization 3*, 1 (2004), 1–18.
13. Inselberg, A. and Dimsdale, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization*, (1990), 361–378.
14. Isenberg, P., Tang, A. and Carpendale, S. An exploratory study of visual information analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (2008), 1217–1226; http://doi.acm.org/10.1145/1357054.1357245.
15. Perer, A. and Shneiderman, B. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. *Proceedings of Intelligent User Interfaces*, (2008), 109–118; http://doi.acm.org/10.1145/1378773.1378788.
16. Segel, E. and Heer, J. Narrative visualization: telling stories with data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1139–1148.
17. Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (1994), 70–77.
18. Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, 1996; http://portal.acm.org/citation.cfm?id=832277.834354.
19. Stolte, C., Tang, D. and Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics 8* (2002), 52–65.
20. Tufte, E. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.
21. van Ham, F. and Perer, A. Search, show context, expand on demand: supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 953–960; http://dx.doi.org/10.1109/TVCG.2009.108.
22. Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J. and McKeon, M. Many Eyes: a site for visualization at Internet scale. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1121–1128.
23. Weaver, C. E. Building highly-coordinated visualizations in Improvise. In *Proceedings of the IEEE Information Visualization Conference*, (2004), 159–166.
24. Wilkinson, L. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Secaucus, NJ, 2005.

Jeffrey Heer (jheer@cs.stanford.edu) is an assistant professor of computer science at Stanford University, where he works on human-computer interaction, visualization, and social computing. In 2009, he was named to *MIT Technology Review*'s TR35 (35 innovators under the age of 35).

Ben Shneiderman (ben@cs.umd.edu) is a professor in the department of computer science, founding director of the Human-Computer Interaction Laboratory, and a member of the Institute for Advanced Computer Studies at the University of Maryland, College Park.