Extreme Visualization:
Squeezing a Billion Records into a Million Pixels

Ben Shneiderman
Human-Computer Interaction Lab & Department of Computer Science
University of Maryland
College Park, MD 20742
ben@cs.umd.edu

ABSTRACT

Database searches are usually performed with query languages and form fill in templates, with results displayed in tabular lists. However, excitement is building around dynamic queries sliders and other graphical selectors for query specification, with results displayed by information visualization techniques. These filtering techniques have proven to be effective for many tasks in which visual presentations enable discovery of relationships, clusters, outliers, gaps, and other patterns. Scaling visual presentations from millions to billions of records will require collaborative research efforts in information visualization and database management to enable rapid aggregation, meaningful coordinated windows, and effective summary graphics. This paper describes current and proposed solutions (atomic, aggregated, and density plots) that facilitate sense-making for interactive visual exploration of billion record data sets.

ACM Classification Keywords
H.5. Information interfaces and presentation (e.g., HCI). H.2 DATABASE MANAGEMENT

General Terms: Human Factors

Author Keywords: Information visualization, database search, dynamic queries, user interface, aggregation, density plots, coordinated windows

INTRODUCTION

The appeal of information visualization is to gain a deeper understanding of a important phenomena that are represented in a database [7]. Of course measuring understanding, comprehension, or knowledge is difficult, but we can study human performance in the process of making known-item searches, information seeking inquiries, and insight discovery events [29, 30].

The tools that support search, browsing, and visualization have dramatically improved in the past decade, so there is value for the database community to re-examine recent work and consider what future opportunities there are for integration of database technologies with interactive information visualization [39].

As one of my professors, Turing award-winner Richard Hamming, wrote: “The purpose of computing is insight, not numbers.” I might paraphrase with “The purpose of visualization is insight, not pictures.” Eye-catching animations, colorful 3D movies, and aesthetic presentations all have a role, but the heart of information visualization is the well-designed user control panel and interaction techniques that enable users to generate task-related comprehensible coordinated windows (selections in one window produce highlighting or new contents in related windows). The successful tools support a process of information-seeking that leads to important insights for individual users, organizational teams, and larger communities. Insights are most valuable if they contribute to solving significant problems in areas such as genomic, scientific, financial, social, economic, political data analysis. The term *insights* makes clear that we are discussing a human experience, made possible by well-designed tools that support discovery. This paper presents the potential for scalable visualizations that use atomic representations, aggregations, and density plots. The examples shown deal with million record databases, and sometimes small ones, but they have the potential for scaling up to a billion records.

The challenge and opportunity for the database community is to develop compact data structures that support algorithms for rapid data filtering, aggregation, and display rendering. If these goals can be achieved while supporting cognitively comprehensible displays with predictable interaction controls, then greatly expanded user communities will be able to explore vast databases. Successful examples with million record databases give hope that academic researchers and industrial implementers can push forward to cope with billion record databases [9, 10, 22].

Most computer users are familiar with *geographic visualizations*, which are typically two-dimensional (2D), even when showing the surface of the earth. They are designed to help users answer questions of adjacency, paths to a destination, and locations of features, as described by east-west and north-south axes. Most computer users are also familiar with common *scientific visualizations*, which show three-dimensional (3D) phenomena often in animated presentations, sometimes user-controlled. These include simulations of storms, airflow over aircraft wings, molecular models, or medical imagery, which are mainly designed to help answer questions of location, such as where the storm intensity is greatest or cancerous lesions are predominant. Relationships such as up-down, left-right, inside-outside are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGMOD'08, June 9–12, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-102-6/08/06...$5.00

important to these users. An especially interesting form of scientific visualization is medical visualizations which typically show 2D images of the human body, although 3D xrays and CAT scans are increasingly used.

Information visualizations are different from geographic and scientific visualizations since they have no inherent 2D or 3D structure, but are designed to deal with *multi-dimensional* and more importantly *multi-variate* data. The attributes for a film database could be as diverse as year of release (integer), genre (categorical), length (real), and leading actor and actress (nominal). Often information visualizations deal with even richer data types such as *time series* of weekly film sales, a *tree structure* of thematic topics, and a *network structure* among actors. The four data types (multi-variate, time series, tree, and network) are tied to tasks such as finding clusters, gaps, outliers, trends, and relationships.

Many visualizations follow The Visual Information Seeking Mantra "Overview first, zoom and filter, then details on demand" [34], which remains effective advice, but an update may be necessary to accommodate a billion records. Overviews remain important to orient users about the extent of the database, distributions of values, gaps in the data, and outliers. Billion record overviews will have to use small atomic markers, larger aggregate markers, or cloud-like density plots. Zooming in on areas of interest and filtering out uninteresting markers helps narrow attention to relevant records. Then users can study individual records and groups to understand them better. With billion record databases zooming and filtering will need to be enhanced by use of aggregate markers that represent hundreds or thousands of atomic markers.

An important contribution from the database community will be to develop scalable data structures and algorithms that support rapid update of visual displays for billion record databases. The successful visualization tools apply carefully designed data structures that run in the high speed store (RAM), so that even users of laptops with a few gigabytes of RAM can interactively explore million record databases. Billion record databases will require compression strategies or innovative hierarchical data management to move data from hard disks to RAM rapidly. While a user may wait several seconds for an aggregation or density plot to be performed, they will subsequently expect interactive performance (approximately 10 frame per second updates) when filtering, smooth zooming, and quick updates to coordinated windows. Precomputing of anticipated data needs can dramatically improve the user experience. In summary, the problems to be overcome include:

- database performance during exploration
- display performance to ensure 100msec updates
- visual representations that are compact and information abundant
- human perception of rich displays with specialized markers, aggregation icons, and density plots
- cognitively comprehensible interaction controls and coordinated windows

ATOMIC VISUALIZATIONS: ONE MARKER PER DATA RECORD

The basic visualizations, such as histograms, time series plots, and two-dimensional scattergrams, show one marker for each data record. These visualizations become more useful when users control the display with innovative widgets such as dynamic queries sliders to select subsets from large databases [2, 3, 33] and zooming to see more details in a specific area [5].

Having double-box dynamic queries sliders to set ranges for integer and real attributes enables users to filter out unwanted items and narrow the display to their interests. Dynamic queries can also be accomplished by item sliders (alphasliders) that allow rapid selection and sweeping through categorical or nominal variables. Check boxes and radio buttons allow AND and OR selections for the values of an attribute. As sliders are adjusted and buttons selected display updates should happen within 100 milliseconds to preserve the cause-effect experience that enhances the capacity for rapid exploration. With these strategies users avoid issuing zero-hit or mega-hit queries and quickly converge on a desired set of records.

Dynamic queries are commonly applied to the basic visualization, but they also apply to the richer data types. Strategies for dealing with multi-variate data include:

- packing more dimensions into a scattergram by using size, color, shape, or rotation for markers
- using multiple 2D scattergrams, usually in a lower triangular matrix
- parallel coordinates to show dimensions simultaneously. Each point in n-space becomes a polyline connecting a point on each of the n-parallel axes [16]
- glyphs, Chernoff faces, and other iconic representations

Popular information visualization strategies for dealing with tree-structured data include:

- node-link diagrams
- treemaps
- hyperbolic trees [23]
- nested indented text.

Finally, strategies for dealing with network data include:

- node-link diagrams with many layout strategies
- adjacency matrices [1, 11].

These basic and richer strategies are effective in showing databases with thousands or up to about a million points using a typical display with 1600 x 1200 pixels. For larger databases there may be overlap, until users zoom in on densely packed areas. This strategy is used in commercial tools and works well up to a few million records.

In the extreme case each database record is mapped to a single pixel, where color indicates the attribute value (Figure 1) [17, 18, 19]. For example, if the records are ordered in useful ways such as by patient age (in upper left square, starting at the upper left and spiraling into the center). If the colors indicate days of hospitalization during the past year, then it is possible to discern a that younger patients have fewer hospitalization days (brown outside) and older patients (yellow in the center) have more. The other squares might show other variables, such as office visits, medications, etc.

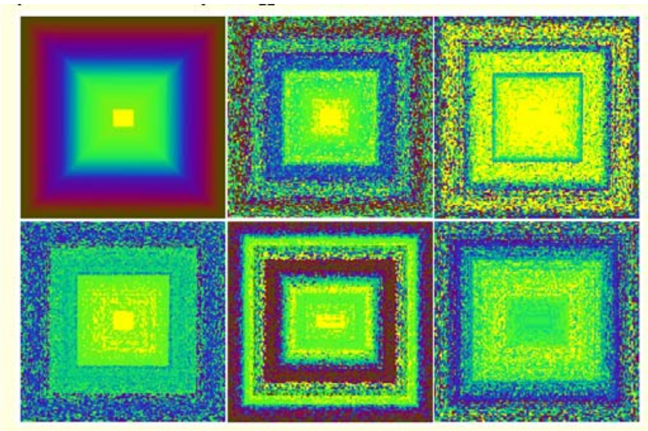


Figure 1: Pixel-based representation of database with six attributes per record. Records are arranged in a square spiral showing relationships among variables.

Similarly, it is possible to pack 1,000 time series each having 1000 time points into a single display and allow zooming to study dense areas in detail. Another approach is to filter out some time series based on their attributes, for example only show the auction bid histories for items that are antique furniture or are sold by a certain seller. Moving from a million to a billion points seems possible but will take some programming to ensure rapid updates.

For tree-structured data, node-link diagrams and hyperbolic trees can support a million nodes if zooming is allowed or special algorithms are used to limit drawing of lower level nodes till users have selected a branch. Treemaps can also accommodate million node hierarchies in a single display (Figure 2) [12].

Million node networks are more difficult to draw within a single display, but coarse representations allow users to see clusters, compare their sizes, and understand their connectivity. Then they can zoom in on areas of interest. Another approach to dealing with large trees is to collapse parts in an accordion-like way (also called *rubber sheet* or *context+focus*), enabling users to smoothly expand regions of interest to show more detail. These techniques have already supported exploration of half million node trees on laptop displays, but moving to larger structures while maintaining animated expansion will take further work [25].

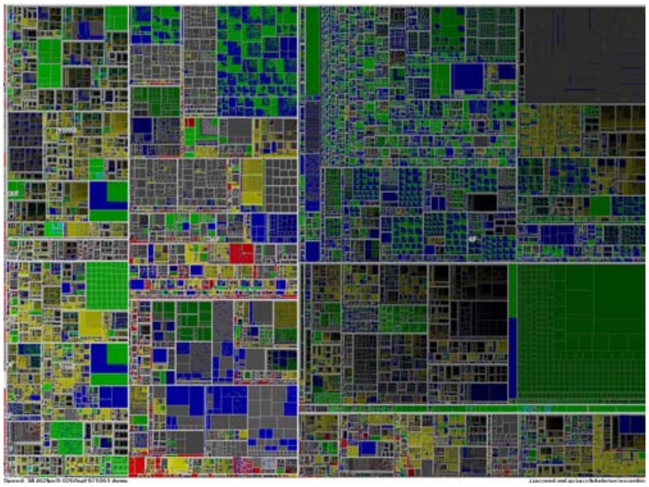


Figure 2: Million node treemap showing the directory structure on a file server. Color encodes file time, area encodes file size.

The success story of the past decade is that these tasks have been successfully implemented in research systems and in a growing array of commercial products, such as Spotfire and Tableau. These products handle at least a million records, provide dynamic query filtering and redisplay at interactive rates so as to support rapid exploration. Their further successes are to import varied data types (integer, real, string, date, time, money, etc.) from traditional relational databases or spreadsheets, provide user control (plus legends) over size, color, shape, and rotation, and allow rapid switching among representation strategies. These commercial tools also support many features such as multiple coordinated windows, data editing, history-keeping, export, report generation, collaboration support, etc.

AGGREGATE VISUALIZATIONS: ONE MARKER PER THOUSAND DATA RECORDS

A natural next step is to push forward from a million records to a billion records. Some researchers are pursuing this goal by moving from mega-pixel to giga-pixel displays. This is often accomplished by tiling 50+ flat-panel monitors to produce wall-sized displays with high resolution [45]. A single computer handles user interaction sending commands to multiple computers which drive the displays. This brute force approach to presenting a billion records has some attraction, but it is difficult to see the whole image and also identify individual pixels.

The more attractive route to seeing a billion records is to find ways to squeeze the information into a million pixels and view it on a commonly available display. The previous section already alluded to two common strategies that involve user control: filtering to see only a subset of the database and coarse views followed by zooming.

This section expands on these ideas by analyzing user needs and then suggesting novel ways to aggregate data in what we might call *aggregate visualizations* [38]. For some tasks, atomic visualizations are necessary, but for many tasks, the aggregate displays are more useful and meaningful. Often clicking on an aggregation marker will cause an expansion in place, but more effectively it will display its components in a coordinated window. The coordinated window could have an entirely different presentation and could contain thousands of records. This hierarchical approach enables scalable solutions for billions of records [41].

The coordinated window approach is well-established in geographic information visualizations, which enable users to study an overview map, then select a region which is shown in detail in a coordinated window. This strategy fits well for database exploration where record attributes enable convenient aggregation. A document database can be explored by seeing an overview by year and topic (Figure 5). Then clicking on a grid cell produces a list of document titles in the upper right view. Clicking on a document title produces a full description of the document in the lower right view [35]. A similar strategy is being added to the commercial tool Spotfire that already supports visualizations with millions of markers. The new feature enables users to select markers to initiate "on-demand" database retrieval that displays in coordinated windows (Figure 6).

An alternate approach for making large databases comprehensible is to compress data with statistical summaries [28] or to convert to linguistic summaries [36]. These methods could be combined with visual representations to present arbitrarily large databases in compact ways

Multivariate Databases

Imagine seeing a crowd of a million people as they assemble for a political rally or music festival. You could fly above to see the entire crowd and understand where the center and periphery of the action are. However, trying to determine what fraction was male/female or the distribution of ages would be difficult. A histogram with age in years on the x-axis with a vertical bar indicating percent of people in each age group would give a quick understanding and allow comparison of ages for a rock concert or a political rally.

But age is only one attribute of individuals, so one histogram is needed for each attribute in a multi-variate database. Fortunately, histograms are effective for most data types, from binary (male/female, YES/NO) to categorical variables (drama/action/mystery/etc.), although nominal values need conversion into something that lends itself to visual representation. These services are common in many Online Analytic Processing (OLAP) systems such as Hyperion or CrystalReports, but richer visualizations would increase their value [37].

Given a database of *n* records with *k* attributes, understanding the distribution of each attribute including gaps and outliers is a great starting point for analysis [14], but then users will want to understand the *k*(*k*-1)/2 pairwise relationships among attributes. Even with a billion records there may only be 10-100 attributes, so the number of pairs is manageable and independent of *n*. A starting point is to look at the linear correlation coefficient between all pairs to understand if one attribute is a simple transform of another, such as product prices in dollars and euros. Analysts will be eager to confirm their knowledge of strong positive linear relationships such as patient height and weight or negative relationships such as county data on unemployment rates and median household incomes. This strategy of ranking strength of features was implemented in the Hierarchical Clustering Explorer (HCE) [31, 32] (Figure 3).

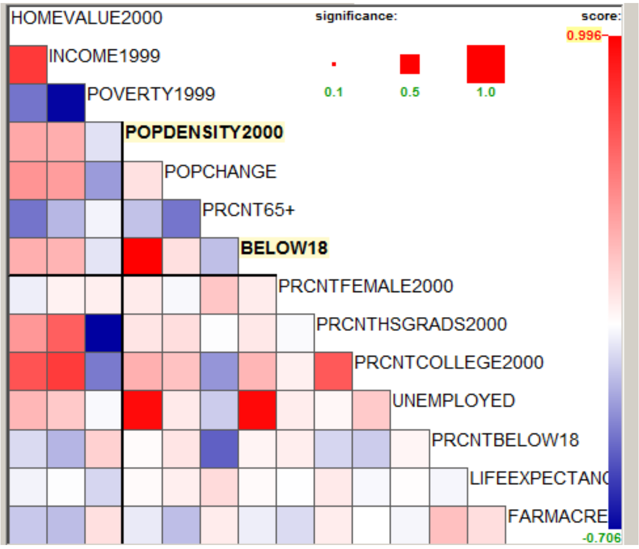


Figure 3: Rank by Feature Framework from Hierarchical Clustering Explorer [31, 32] uses red to highlight the strong positive linear correlations for 14 attributes of 3128 U. S. counties, the strongest being Population Density and Below18 Population. The size of the lower triangular matrix (for this county database of 14 attributes it has 91 cells) is independent of number of records, but reveals much about the relationship among variables (http://www.cs.umd.edu/hcil/hce).

Quadratic, sinusoidal, or exponential relationships are also of interest for each of the *k*(*k*-1)/2 pairs of attributes. The algorithms for carrying these out are scalable up to a billion records.

Outliers, clusters, and gaps in two and higher dimensional visualizations are of great interest and can be highlighted for users to explore. There are a variety of outlier and cluster algorithms, but very few gap detection algorithms, yet these were very informative in our case studies with users [31]. Outlier detection algorithms can be scaled to a billion items but clustering and gap detection are problematic, so much work remains to be done and before even considering how to display the results. This idea was called scattergram diagnostics, or *scagnostics*, by the famed statistician John Tukey in a 1985 speech [40], but we believe that the Hierarchical Clustering Explorer is the first implementation of this idea. Proposals for further scagnostics were made by Wilkinson and his colleagues [43].

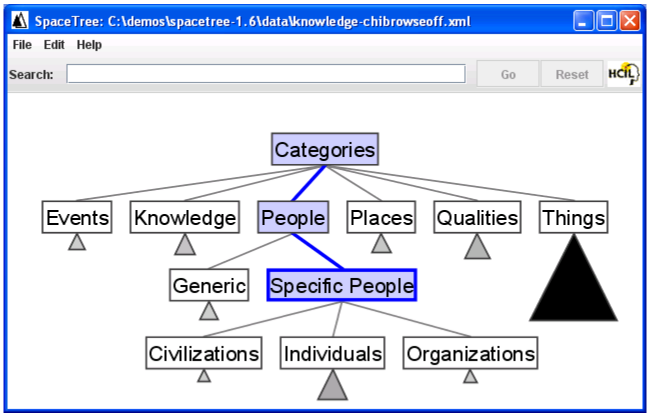


Figure 4: Spacetree, a scalable solution for large trees, shows a partially expanded tree with aggregation markers that are user expandable (http://www.cs.umd.edu/hcil/spacetree).

Time Series

Time series databases can grow large in the number of time points or the number of time series. Environmental sensors may capture temperature, humidity, barometric pressure, etc. every minute for a year yielding 525,600 data points for each variable. With only a thousand sensors, there are well over a billion data points. Seeing even one time series would require aggregation to fit on a 1600-pixel wide display, but the methods are well understood [6, 15]. Seeing seasonal patterns will be easy, but sharp rises/falls over a few hours will be difficult. Understanding spatial relationships such as temperature shifts with altitude in mountainous areas or changes according to prevailing west-to-east winds will be more difficult and require a coordinated geographic view.

Sometimes, the number of time series is large, such as if the thousand environmental sensor streams were broken into 365,000 daily time series of 1440 time points each. Each time series fits on the display, but it is difficult to see any individual time series. Here aggregation by sensor or month would be useful, as would clustering into a smaller number of closely related *meta-time-series*.

Tree Structures

The ubiquity of hierarchies makes tree structures an important data type and the one with the most varied visualizations. Tree structures are conveniently organized for aggregation since first few levels of the tree are a natural representation of the full tree. For example, a major legal database has 100+ million documents classified into a 85,000 node tree with 23 levels. However, frequent users are familiar with the first three levels of the tree that consist of less than a thousand nodes. Showing a node-link diagram of the 3-level tree is feasible with size or color coding to indicate how many documents are available in the sub-trees.

A natural approach is to give user control over which nodes are exposed, which is the strategy in SpaceTree (Figure 4) [27]. Initially, the root and first level nodes are shown. Users can open lower levels on demand. The darkness of the triangles hanging from each node indicate the total number of nodes below, and the height indicates the number of levels. A similar solution was offered in DOITree [8].

Networks

Drawing large networks (million nodes and more) is such a challenge that there is an annual Graph Drawing conference devoted to this problem. Recent breakthroughs have enabled million node visualizations to be drawn in a few seconds [20, 21]. However, the dynamic queries that users have come to expect are more difficult to arrange on these visualizations, because redrawing with incremental changes is difficult in the force-directed approaches that are commonly used. Furthermore, scaling to a billion nodes will take some innovative thinking. An alternative is to draw coarse views of the network so users can see the main clusters, compare their sizes, and understand the connectedness among clusters (Figure 7) [44]. A related approach, used in SocialAction, is to compute community structures and then represent the communities by a single aggregate node (Figure 8) [26]. The current version handles 150,000 node networks, but scaling up to a billion nodes will require improved techniques**.**

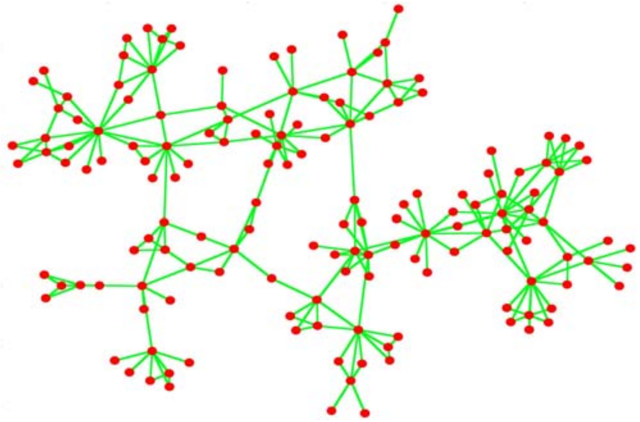
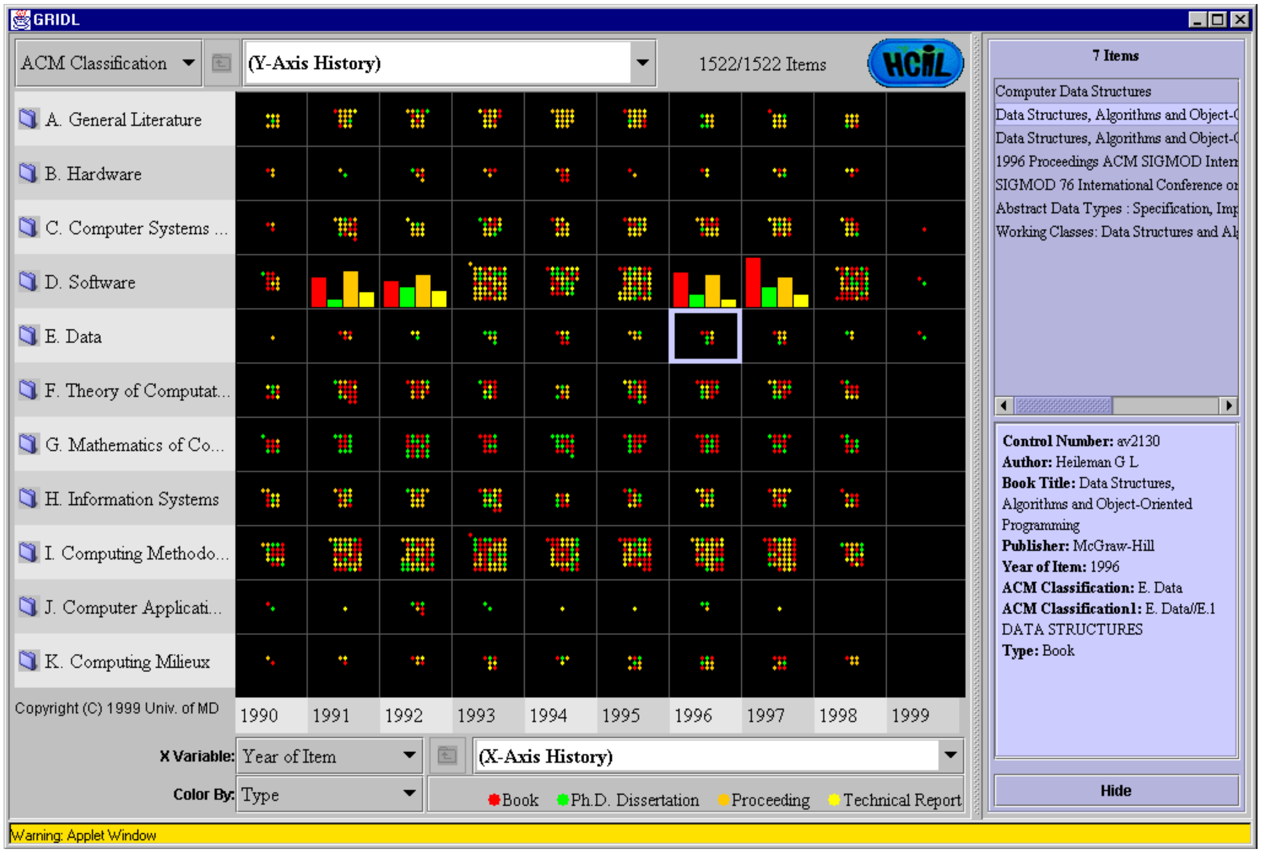


Figure 7: This coarsened network of 152 nodes represents a larger network with 46,480 nodes. The authors's approach in GreenMax [44] has been tested in million node networks and the authors claim to be able to scale to larger networks.

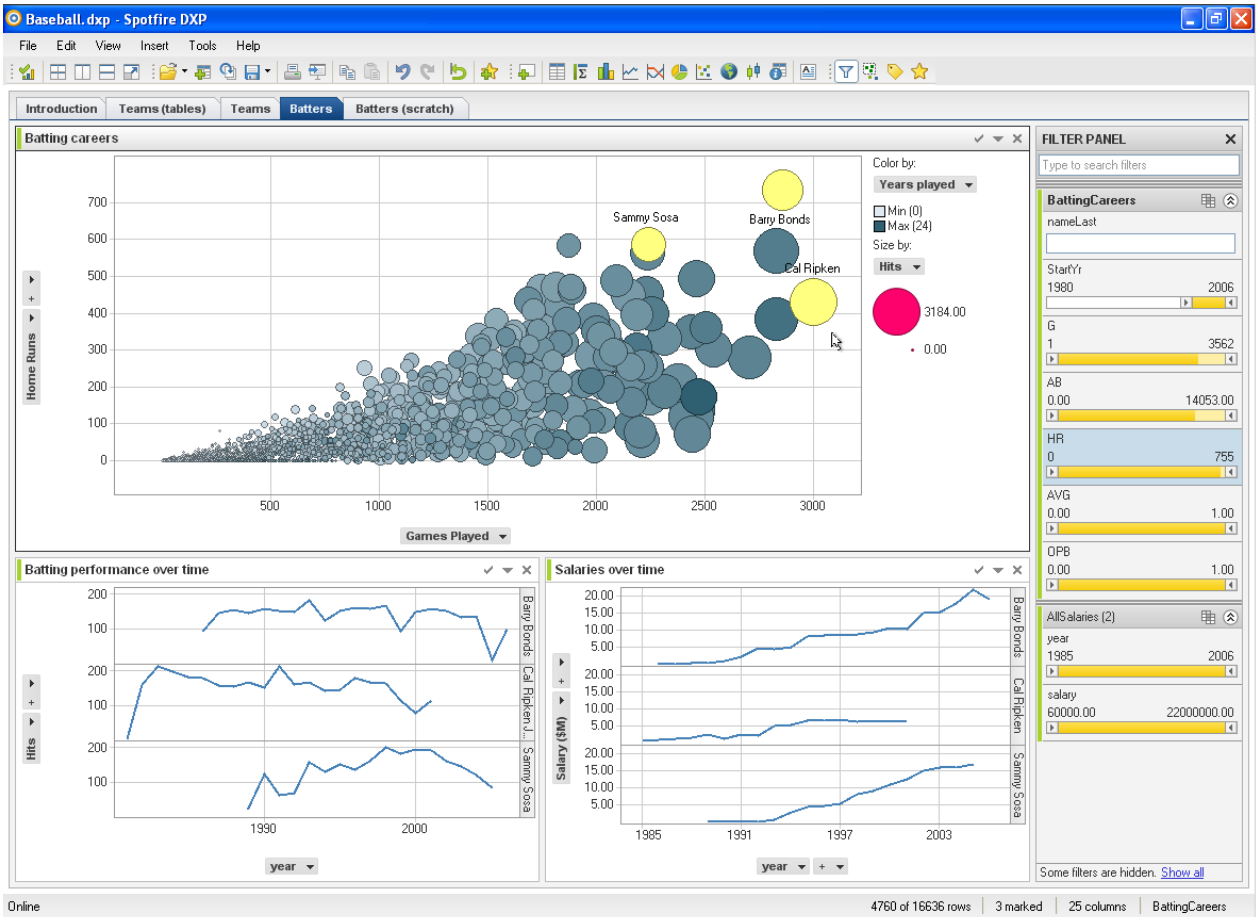
Figure 5 (next page, top): This Graphical Interface for Digital Libraries (GRIDL, www.cs.umd.edu/hcil/west-legal/gridl/) offers a scalable approach where each axis is an expandable hierarchy. Each grid cell shows up to 49 colored dots for documents, and shifts to an aggregation marker in the form of a bar chart to show the relative proportions of each document type. Clicking on a grid cell produces a listing of titles in the upper right window. Clicking on a title produces the catalog description in the bottom right window.

Figure 6 (next page, bottom): Spotfire's new On-Demand feature enables dynamic data retrieval from large databases when needed. In the scattergram, which shows games played vs home runs hit, each circle indicates a player (size is number of hits), which acts as an aggregate marker for their career data. Three players have been selected (Bonds, Ripken & Sosa) triggering a database access to display the lower time series showing their career time series of hits and salary.



Computer Data Structures
Data Structures, Algorithms and Object-(
Data Structures, Algorithms and Object-(
1996 Proceedings ACM SIGMOD Intern
SIGMOD 76 International Conference o
Abstract Data Types : Specification, Imp
Working Classes: Data Structures and Al

Control Number: av2130
Author: Heileman G L
Book Title: Data Structures, Algorithms and Object-Oriented Programming
Publisher: McGraw-Hill
Year of Item: 1996
ACM Classification: E. Data
ACM Classification1: E. Data/E.1 DATA STRUCTURES
Type: Book



Online 4760 of 16636 rows 3 marked 25 columns BattingCareers

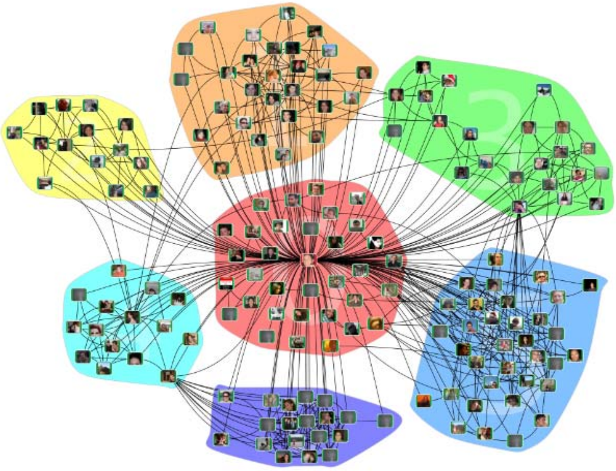


Figure 8: Grouping nodes into community structures based on link relationships helps bring order to a small Facebook social network. In this small example, each community can be replaced by a single aggregate node, enabling scaling up to large databases (http://www.cs.umd.edu/hcil/socialaction).

By contrast the novel approach of using a semantic substrate to lay out the nodes in a 2D grid plot is scalable, since the contents of each grid cell can be represented by a metanode whose size is proportional to the number of nodes in that cell [4]. A semantic substrate consists of a set or rectangular regions in which nodes are places according to node attribute values. Each region is similar to a 2D scattergram and there for the multivariate data techniques described earlier can be applied.

Links are drawn only on user request with a control panel that allows selective drawing of links to minimize clutter. This filtering approach can ensure that links are drawn only between nodes in a single region or only connecting a pair of regions. The NVSS implementation of semantic substrates (http://www.cs.umd.edu/hcil/nvss) used a gridded scatterplot strategy, much like GRIDL, which lends itself conveniently to replacing all the nodes in a grid cell with a single metanode (see transition from Figure 9 to Figure 10). The reduction in nodes enables users to see distributions and greatly clarifies link visibility so users can follow links from source to destination more often.

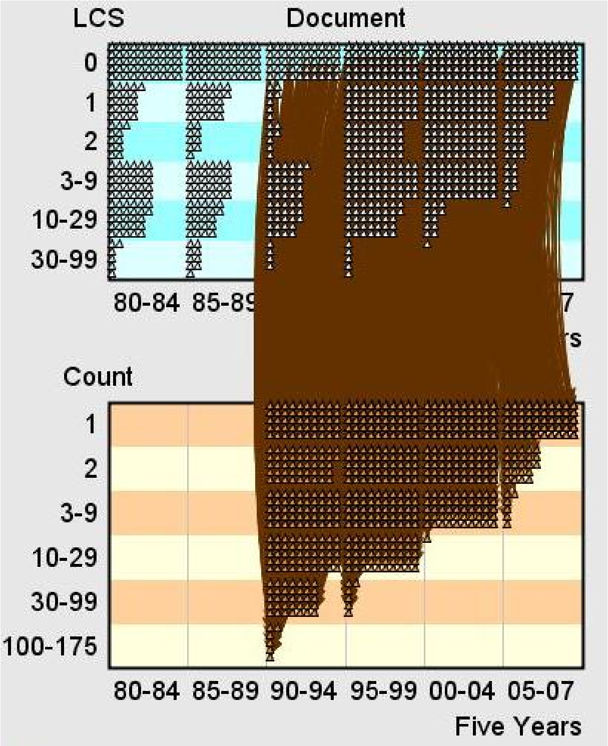


Figure 9: A semantic substrate with the upper region for 1700 Documents and the lower region for 2567 Keywords that are used for this topic (both shown as triangles). Documents are organized by year and Keywords by the year they were first used. Count indicates the number of times a keyword was used. Documents are linked to keywords they use. The overplotting of nodes and 9649 links means this diagram has little utility.

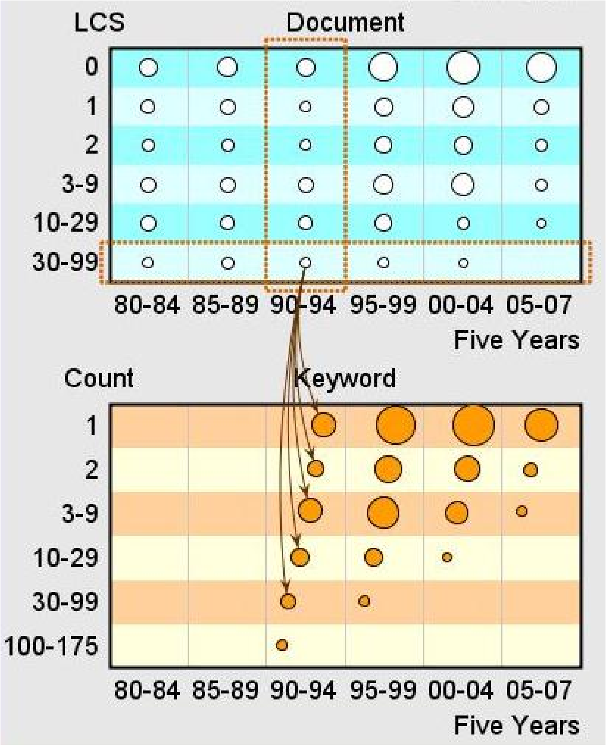


Figure 10: The same semantic substrate as Figure 9, but metanodes (circles) have replaced the nodes in each cell. Metanode sizes indicate the number of nodes they replace. With aggregate nodes and fewer links, plus filters on the outgoing links, the visualization allows relationships to be seen. One surprise for our domain expert partners was the absence of a link to the most frequent keywords (metanode in the 90-94 cell). Figures 9 and 10 were created using the Network Visualization with Semantic Substrates (NVSS) (http://www.cs.umd.edu/hcil/nvss).

DENSITY PLOT VISUALIZATIONS: COLOR CODED AREAS SHOW USERS WHERE TO EXPLORE

When individual markers representing individual records fill a visualization, clustering strategies are useful to organize them into aggregate markers. A special form of aggregation is the density plot which uses a spatial substrate organizing principle, but shows concentrations of markers. This could be interpreted as a two-dimensional histogram (Figure 11).

For multivariate data, two-dimensional projections with scattergrams are commonly used, but three-dimensional density plots have been developed.

For time series data, density plots can show concentrations of time points. A good model is the work on cluster displays in parallel coordinate views (Figure 12) [13].

For tree structures presented in node-link diagrams a density plot seems viable. Figure 13 presents a mockup of what a 5-level density plot might look like. This density plot can accommodate trees with arbitrarily large fan-out and depth. The sum of the densities at each level is 100%. Each cell indicates percent of the nodes in the subtrees below.

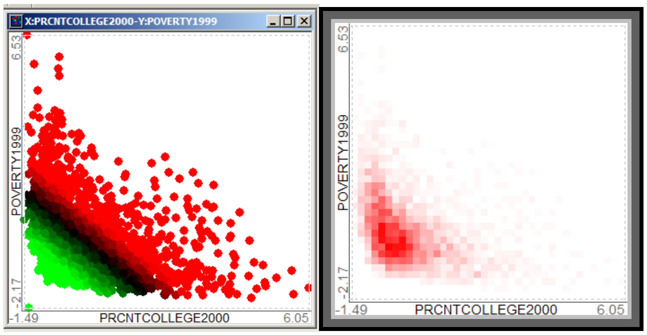


Figure 11: The scattergram from HCE (on the left) is heavily overplotted, but converting to a 40 by 40 grid plot (on the right), enables users to see the distribution density. Clicking on a grid cell brings up the records in that cell.

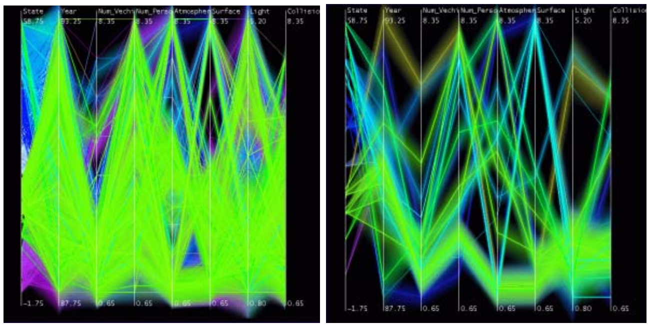


Figure 12: Parallel coordinate shows 230,000 records in a fatal accident database on the left. The variable opacity bands show meaningful clusters on the right.

Treemaps can also show density by aggregating subtree counts or attributes. Existing tools such as Treemap 4.0 (http://www.cs.umd.edu/hcil/treemap) allow a color coded density plot that shows the number of nodes or aggregate values of node attributes. The user interface has a slider for level, so aggregations can be made dynamically for any level of the tree. Figure 14 shows a three-level summary of a 23 level tree. Rendering only the first few levels of a tree makes this approach potentially scalable to a billion nodes.

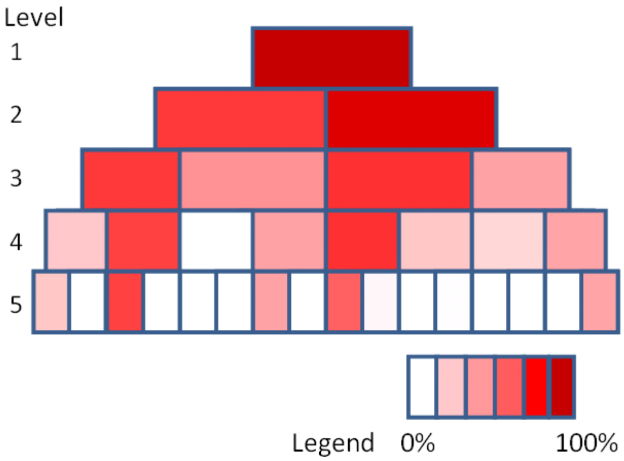


Figure 13: Tree density plot showing percentage of nodes in subtrees at each level. This mockup is designed to accommodate arbitrary fanout (medial split to left and right at each level) and arbitrary depth (level 5 summarizes lower levels). It shows that 2 of the 16 subtrees at level 5 contain most of the remaining nodes.

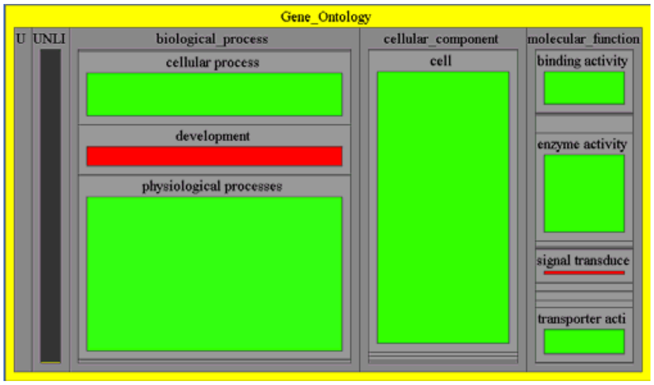


Figure 14: A treemap density plot showing a 3-level summary of gene expression data for a 23-level tree with 22,995 nodes. Red areas indicate high activity, green low activity. This treemap was generated using the gene ontology and gene expression data (http://www.cs.umd.edu/hcil/treemap). Red areas show high activity in the development subtree of the biological processes and signal transducer subtree of molecular function.

CONCLUSION

The benefits of visual exploration are increasingly well understood, raising expectations of users who want to explore ever larger databases. Gigapixel displays will be useful for some tasks, but innovative interface design is likely to have higher payoffs and wider usage. Current atomic visualizations build on pixel-based representations, filtering to show subsets, and zooming to focus on areas of interest. Meaningful aggregate visualizations show the greatest promise because they promote sense-making while keeping display complexity low. Aggregation markers, which can represent thousands of records, can be organized and presented so as to suggest where users should click. When they click, the aggregation markers can open in place or present their contents in a coordinated window that might have an entirely different representation. Density plots offer some fresh possibilities, especially for statistically minded users. It seems that databases systems will follow the path of operating systems. Most operating systems users have shifted from command line interfaces to graphical user interfaces, greatly expanding the audience for computing. Similarly, the narrow community of database query language users will expand greatly as effective visualization interfaces enable rapid and comprehensible access to large databases. If strong collaborations can be arranged between information visualization and database management researchers and implementers, then the use of billion record visualizations could become widespread.

ACKNOWLEDGMENTS

Thanks to Dennis Shasha and the SIGMOD 2008 program committee for inviting this keynote. Thanks also to Benjamin Bederson, Amol Deshpande, Jean-Daniel Fekete, Francois Guimbretiere, Adam Perer, Hanan Samet, and Dennis Shasha for helpful comments on drafts.

REFERENCES

1. Abello, J., van Ham, F., and Krishnan, N., ASK-GraphView: A Large Scale Graph Visualization System, *IEEE Trans. Visualization & Computer Graphics 12*, 5 (2006), 669-676.
2. Ahlberg, C. and Shneiderman, B., Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Display. *Conference proceedings on Human factors in computing systems*, April 1994, 313–318, ACM New York.
3. Ahlberg, C. and Shneiderman, B., AlphaSlider: A compact and rapid selector, *Proc. of ACM CHI94 Conference*, ACM Press, New York (April 1994), 365-371.
4. Aris, A. and Shneiderman, B., A node aggregation to reduce complexity in network visualizations using semantic substrates, University of Maryland Technical Report, Dept of Computer Science (February 2008).
5. Bederson, B. B., & Meyer, J., Implementing a Zooming User Interface: Experience Building Pad++, *Software: Practice and Experience, 28,* 10 (1998), 1101-1135.
6. Buono, P., Aris, A., Plaisant, C., Khella, A., and Shneiderman, B., Interactive pattern search in time series, *Proc. SPIE Conference on Visual Data Analysis*, SPIE, Washington, DC (January 2005), 175-186.
7. Card, S. K., MacKinlay, J. D., Shneiderman, B., *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publishers, San Francisco, CA (1999).
8. Card, S. K. and Nation, D., Degree of Interest Trees: A Component of an Attention-Reactive User Interface, *Proc. Advanced Visual Interfaces,* Available from ACM Press, New York (2002).
9. Chen, C., Top 10 Unsolved Information Visualization Problems, *IEEE Computer Graphics & Applications* (July/August 2005), 12-16.
10. Eick, S. and Karr, A., Visual Scalability, *Journal of Computational and Graphical Statistics 11*, 1 (March 2002), 22-43.
11. Elmqvist, N., Do, T.-N., Goodell, H., Henry, N., and Fekete, J.-D., ZAME: Interactive Large-Scale Graph Visualization, *Proc. IEEE Pacific Visualization Symposium 2008*, IEEE Press (March 2008), 215–222.
12. Fekete, J.-D., Plaisant, C., Interactive Information Visualization of a Million Items, *Proc. IEEE Symposium on Information Visualization 2002* (InfoVis 2002, Boston, USA), IEEE Press, Los Alamitos, CA (October 2002), 117-124.
13. Fua, Y.-H., Ward, M., and Rundensteiner, E., Hierarchical Parallel Coordinates for Exploration of Large Datasets, *Proc. IEEE Visualization'99*, IEEE Press, Los Alamitos, CA (1999), 43-50.
14. Guha, S., Koudas, N., and Srivastava, D., Fast algorithms for hierarchical range histogram construction, *Proc. ACM Symposium on Principles of Database Systems*, ACM Press, New York (2002), 180-187.
15. Hochheiser, H. and Shneiderman, B., Dynamic query tools for time series data sets, Timebox widgets for interactive exploration, *Information Visualization 3,* 1 (March 2004), 1-18.
16. Inselberg, A. and Dimsdale, B., Parallel coordinates: A tool for visualizing multidimensional geometry, *Proc. Visualization'90* (San Francisco, Oct. 23–26). IEEE Press, Los Alamitos, CA (1990), 361–370.
17. Keim, D. A., Information Visualization and Visual Data Mining, *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (January 2002), 1-8.
18. Keim, D. A., Visual Exploration of Large Data Sets, *Communications of the ACM* 44, 8 (August 2001), 38-44.
19. Keim, D. A., Designing Pixel-Oriented Visualization Techniques: Theory and Applications, *IEEE Trans. on Visualization and Computer Graphics* 6, 1 (January 2000), 59-78. [doi>10.1109/2945.841121]
20. Koren, Y., Carmel, L., and Harel, D., Drawing Huge Graphs by Algebraic Multigrid Optimization, *SIAM Multiscale Modeling and Simulation* 1, 4 (2003), 645-673.
21. Koren, Y., Carmel, L., and Harel, D., ACE: A Fast Multiscale Eigenvector Computation for Drawing Huge

Graphs, *Proc. IEEE Information Visualization 2002 (InfoVis'02)* 2002), 137-144.

22. Kreuseler, M., Lopez, N., and Schumann, H., A. scalable framework for information visualization, *Proc. IEEE Symposium on Information Visualization* (2000), 27-36.

23. Lamping, J., Rao, R., and Pirolli, P., A focus+context technique based on hyperbolic geometry for visualizing large hierarchies, *Proc. of ACM CHI95 Conference*, ACM Press, New York (1995), 401-408.

24. Munzner, T., Drawing Large Graphs with H3Viewer and Site Manager, *Proc. Symp. Graph Drawing'98* (1998): 384-393.

25. Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., and Zhou, Y., TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility, *ACM Trans. on Graphics 22,* 3 (2002), 453-462.

26. Perer, A. and Shneiderman, B., Balancing systematic and flexible exploration of social networks, *IEEE Symposium on Information Visualization* and *IEEE Transactions on Visualization and Computer Graphics 12,* 5 (October 2006), 693-700.

27. Plaisant, C., Grosjean, J., Bederson, B., SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation, *IEEE Symposium on Information Visualization* (2002), 57 -64.

28. Saint-Paul, R., Raschia, G., and Mouaddib, N., General purpose database summarization, *Proc. 31st International Conference on Very Large Data Bases*, Trondheim, Norway (August 30-September 02, 2005), 733-744.

29. Saraiya, P., North, C., and Duca, K., An Insight-Based Methodology for Evaluating Bioinformatics Visualization, *IEEE Trans. Visualization and Computer Graphics 11*, 4 (July/Aug. 2005).

30. Saraiya, P., North, C., and Duca, K., An Evaluation of Microarray Visualization Tools for Biological Insight", *IEEE Symposium on Information Visualization 2004 (InfoVis 2004)* (2004), 1-8.

31. Seo, J. and Shneiderman, B., Knowledge discovery in high dimensional data: Case studies and a user survey for the rank-by-feature framework, *IEEE Transactions on Visualization and Computer Graphics 12,* 3 (May/June, 2006), 311-322.

32. Seo, J. and Shneiderman, B., A rank-by-feature framework for interactive exploration of multidimensional data, *Information Visualization* 4, 2 (June 2005), 99-113.

33. Shneiderman, B., Dynamic queries for visual information seeking, *IEEE Software*, 11, 6 (1994), 70-77.

34. Shneiderman, B., The eyes have it: A task by data-type taxonomy for information visualizations, *Proc. Visual Languages* (Boulder, CO, Sept. 3–6). IEEE Computer Science Press, Los Alamitos, CA (1996), 336–343.

35. Shneiderman, B., Feldman, D., Rose, A., and Ferre, X. A., Visualizing digital library search results with categorical and hierarchical axes, *Proc. 5th ACM International Conference on Digital Libraries,* ACM, New York (June 2000), 57-66.

36. Sripada, S. G., Reiter, E., Hunter, J., and Yu, J., Generating English Summaries of Time Series Data using the Gricean Maxims, *Proc. ACM Conference on Knowledge Discovery and Data Mining* (KDD) (2003), 187-196.

37. Stolte, C., Tang, D., and Hanrahan, P., Multiscale visualization using data cubes, *Proc. Eighth IEEE Symposium on Information Visualization*, Boston, MA (October 2002), 7-14.

38. Tang, L. and Shneiderman, B., Dynamic aggregation to support pattern discovery: A case study with web logs, *Proc. Discovery Science: 4th International Conference 2001*, Editors (Jantke, K. P. and Shinohara, A.), Springer-Verlag, Berlin (March 2001), 464-469.

39. Thomas, J.J. and Cook, K.A. (eds.), *Illuminating the Path: Research and Development Agenda for Visual Analytics*, IEEE Press (2005).

40. Tukey, J. W. and Tukey P. A., Computer graphics and exploratory data analysis: An introduction. *Annual Conference and Exposition: Computer Graphics 1985* (Fairfax, VA, USA), National Micrographics Association: Silver Spring; 3 (1985), 773–785.

41. Ward, M., Peng, W., and Wang, X., Hierarchical visual data mining for large-scale data, *Computational Statistics* 19 (2004), 147-158.

42. Wattenberg, M., Visual Exploration of Multivariate Graphs, *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, New York (2006), 811-819.

43. Wilkinson, L., Anand, A., and Grossman, R., Graph-theoretic scagnostics, *Proc. IEEE Information Visualization 2005 (INFOVIS'05)* (2005), 157-164.

44. Wong, P. C., Foote, H., Mackey, P., Chin Jr., G., Sofia, H., and Thomas, J., A Dynamic Multiscale Magnifying Tool for Exploring Large Sparse Graphs, *Information Visualization* 7, 2 (June 2008, to appear).

45. Yost, B., Haciahmetoglu, Y., and North, C., Beyond visual acuity: the perceptual scalability of information visualizations for large displays, *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*, San Jose, California, USA (April 28-May 03, 2007), 101-110.