Making Information More Accessible: A Survey of Information Visualization Applications and Techniques

Gary Geisler

Last updated January 31, 1998

In the relatively brief history of computing, the methods by which users interact with digital information have been largely artificial and unintuitive. Although people perceive information in the world through a combination of sight, sound, touch, feel, and smell, people using computers have perceived information primarily through static, monochromatic displays of text and numbers. The development of the NCSA Mosaic Web browser in 1993 made the digital information of the Internet more accessible by incorporating graphics and sound with text and by making navigation easy with a point-and-click interface. The subsequent exponential growth of the World Wide Web is a dramatic demonstration of how information can be made more accessible by incorporating visualization techniques.

Visualization is only one aspect of a broader range of methods of interacting with digital information that we will see in the future. Haptic (touch) feedback is beginning to see use in specialized applications, and the use of audio is becoming more common. People perceive information, however, primarily through vision, and the display of digital information is likely to continue to be designed primarily for visual perception. Although the computer world as a whole has not yet moved too far from monochromatic displays of text and numbers, the emerging field of information visualization has already produced many interesting examples of how information can be made more accessible through visual representations.

This paper surveys the variety of ways visualization is being used to make information more accessible. Most of the applications and techniques discussed in this paper have been in development for less than a decade. Other visualization applications have been in existence for quite a bit longer; scientific visualization, for example, is a well-established field. But scientific visualization is also a relatively specialized field, focused on data that describes physical objects and scientific measurements. It is a way to visualize real objects that are otherwise difficult to see and manipulate, such as molecular structures, or to view simulations of scientific phenomena, such as the flow of air over wings.

Because of this specific focus, scientific visualization has been the domain of a relatively small number of trained scientists.

While developments in scientific visualization are quite interesting, this paper concentrates on information visualization designed for a broader audience. Specifically, this paper surveys visualization techniques and applications designed to enable a wide variety of computer users to more easily navigate information spaces, to better display retrieved information, and to improve their understanding of information.

The Need for Information Visualization

The amount of information available today is greater than ever. In addition to the kinds of data that has been stored on computers for decades, many other types of information – newspapers, magazines, catalogs, procedures, forms, guidelines, live audio and video broadcasts – are being digitized and made available to computer users. People are using larger hard drives and high-capacity storage devices to store more information locally, organizations are using company-wide intranets to store information for employees, and the World Wide Web has put a tremendous amount of information at every network-connected computer user's fingertips.

Information visualization enables people to deal with all of this information by taking advantage of our innate visual perception capabilities. By presenting information more graphically we make it possible for the human brain to use more of its perceptual system in initially processing information, rather than immediately relying entirely on the cognitive system. For instance, imagine the task of determining the high and low prices of a company's stock for a time period, given the following table of dates and corresponding stock prices:

Date	Price	Date	Price
8/1	104 3/4	8/18	104
8/4	106 1/4	8/19	107 15/16
8/5	106 1/2	8/20	108
8/6	107 7/8	8/21	105 3/4
8/8	105 1/4	8/22	106 3/8
8/11	103	8/25	105

To complete the task you scan down the list, mentally noting the extreme values you come across, replacing those values with any higher or lower values, until you reach the end of the list. You then find the dates that correspond with the most extreme values. For a week or a month's worth of data, it is not too difficult a task; as the list gets

8/12	103 7/16	8/26	103 5/16
8/13	104 3/8	8/27	103 5/16
8/14	103 5/8	8/28	101 1/8
8/15	99 15/16	8/29	101 3/8

longer, though, say an entire year, the task becomes more difficult.

Completing the task using a line graph of the company's stock price, however, is much simpler, especially for longer periods of data. One glance reveals the highest and lowest values on the chart.

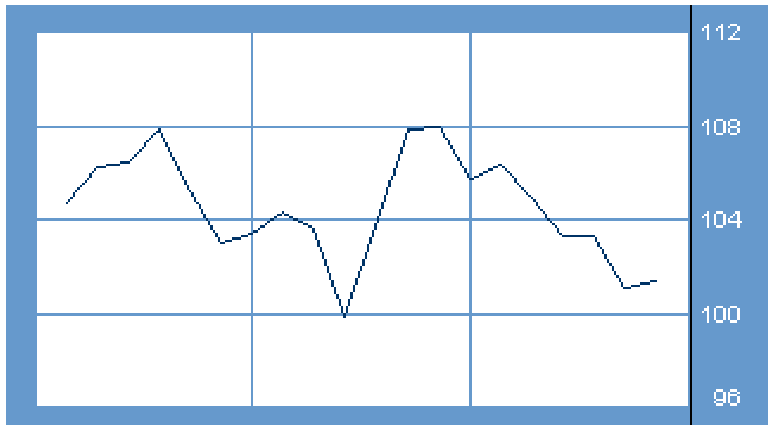


Figure 1. Stock price graph
Source: Stocksite

Information visualization applications rely on basic features that the human perceptual system inherently assimilates very quickly: color, size, shape, proximity, and motion. These features can be used by the designers of information systems to increase the data density of the information displayed (Tufte, 1990). Because we perceive such features so readily, and because each feature can be used to represent different attributes of data, good visualizations enable us to not only perceive information more easily but also to perceive more information at one time. We can immediately see patterns in data that indicate trends, recognize gaps in the data, discover outliers or errors in the data, pinpoint minimum and maximum values, and identify clusters. As a result, information visualization applications enable us to better understand complex systems, make better decisions, and discover information that might otherwise remain unknown.

Information Visualization Applications

Research in information visualization has produced a wide variety of applications. The look and function of an application is determined by the type of user tasks the visualization is designed to support, the kind of data being visualized, and the way the application uses basic features such as color, size, shape, proximity, and motion to represent data. While there are several ways to survey this range of information visualization applications, perhaps the most fundamental method is to categorize them by the underlying data types that the applications attempt to visualize. This organization by data type, as outlined by Ben Shneiderman (Shneiderman, 1998), identifies seven data types:

- One-dimensional
- Two-dimensional
- Three-dimensional
- Multi-dimensional
- Temporal
- Hierarchical
- Network

This paper will survey the information visualization field by describing applications relevant to each data type. Many applications, however, are based on data that can be placed into more than one data type category, making this organization somewhat artificial.

Visualizing One-Dimensional Data

One-dimensional data is simple, linear data, such as text or a list of figures. The text documents, lists of names and addresses, and the source code of computer programs that many people work with daily are based on one-dimensional linear data. The usefulness of visualizing one-dimensional data depends on the size of the data and the tasks the user wants to perform with the data.

Probably the most common form of one-dimensional data is a text document. In most cases, there is no need to use visualization for text documents. They are simply read from start to finish, or specific parts are referred to when necessary. In other cases, however, we can take advantage of the capabilities of computers and digital information and use visualization to increase the usefulness of a one-dimensional text document, enabling the user to see things or make connections between parts of a document that would otherwise be more difficult.

Consider the plays of Shakespeare, for example. Using traditional paper copies of a play, or even a simple digitized version of a play, an actor or scholar can read the text, flip back and forth between sections of interest, and in the case of the digitized version, perhaps even search for specific words or phrases. A project by the Visible Languages Group at MIT, however, uses the plays of Shakespeare to show that there is much more you can do with one-dimensional text using visualization.

Based on methods that enable the … size, … tio … e [Virtual Shakespeare Project](#) … s of … ives … ate and … ew of … ndred thousand words ([Small](#), 1996). At this scale, individual words aren't legible, but patterns and the overall structure of individual plays are clear. As shown in Figure 2, changes in scale enable the viewer to see the relative size and structure of all the Shakespeare plays (view in lower right), the number and relative length of the acts in the plays (lower left), the frequency and length of dialog by characters in a particular play (upper right), and the full, legible dialog of a play (upper left).

Visual filters can be used to discover useful patterns and structures. For instance, all dialog for a specific character can be highlighted and visible from the view at any scale (Figure 3). Through changes in color the viewer can see how many lines a specific character has and how they are distributed throughout the entire play, a capability that would be impossible in traditional book form. Other potentially useful features include unobtrusive displays of footnotes and rapid navigation within a large amount of text.

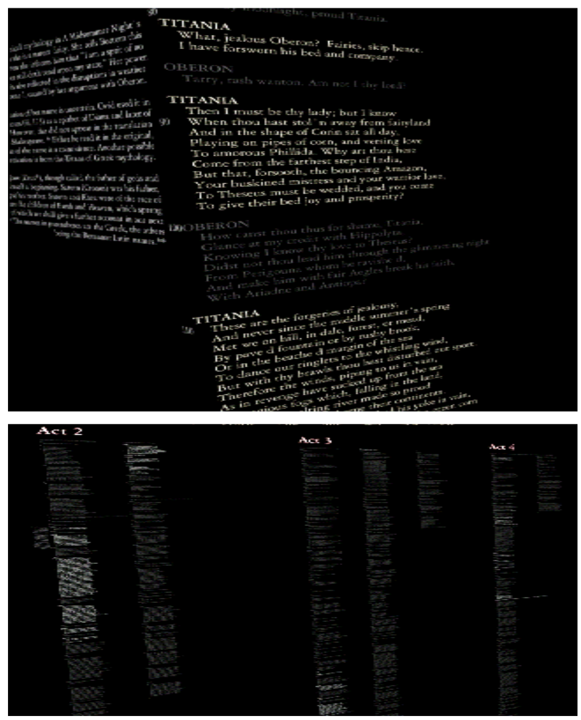


Figure 3. Filtering dialog with color
Source: [IBM Systems Journal](#)

The basic idea of the Virtual Shakespeare Project is applied to software code in the [SeeSoft](#) software visualization system from Lucent Technologies' Bell Labs. This visualization uses color and scaling to enable a user to view a large software program as a whole, as in Figure 4, while also providing access to details, as in Figure 5. The visualization provides a quick means of understanding how the components of a program fit together, which lines of code were most recently edited, and brings together other features that are difficult to comprehend by scrolling through lines of code in a traditional manner.

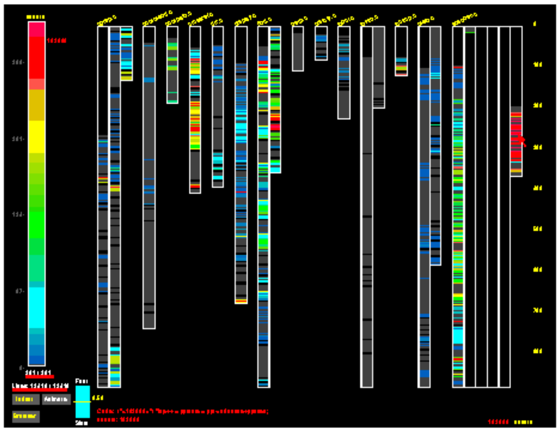


Figure 4. SeeSoft code viewer - overview
Source: [Lucent Technologies]

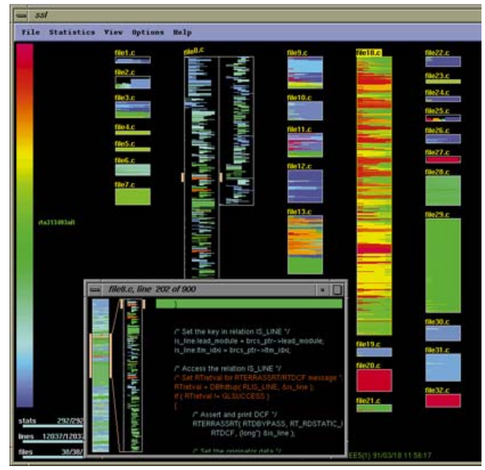


Figure 5. SeeSoft code viewer - c
Source: [Lucent Technologie

Providing access to details while not obscuring the larger picture is a common theme in information visualization. Another example of this principle at work is the TileBars system of showing query results. When you submit a query to a typical web search engine, the results of your query are usually displayed in a series of web pages, each containing a text listing of documents that the system has determined are relevant to your query. Each listing consists of the title of each document listed by title of the web page, URL, and a form of an abstract extracted from the page. But aside from the ordering of the documents, which is usually done according to a system-defined relevance rating, no other information is available from the data.

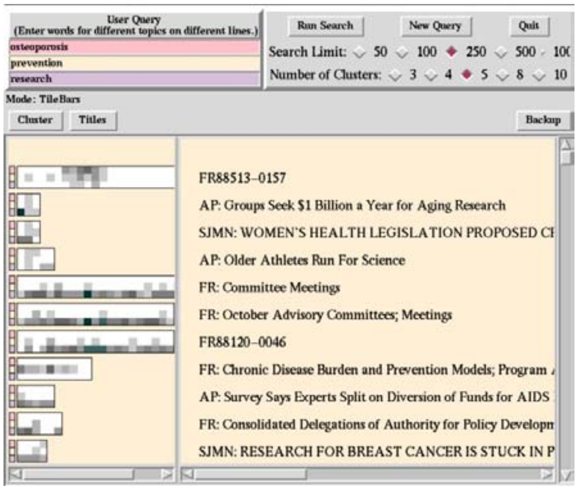


Figure 6. TileBars
Source: Marti Hearst, UC Berkeley

TileBars (Hearst, 1995) provides the same information as in the standard query results but uses visualization to provide additional useful information. Figure 6 shows how TileBars visualizes the query results. In this example, the query terms for the search are osteoporosis, prevention, and research. The long horizontal rectangle to the left of each retrieved document shows the relative size of the document (length of the rectangle) and contains a row for each query term. How each query term is distributed in the document is indicated by shaded squares corresponding to a part of the document. The darkness of the shading represents the frequency that the query term occurs in that part of the document; the darker the square the more often the term occurs in that section.

By visually representing the frequency and distribution of query terms throughout retrieved documents, the TileBars approach provides much more information in the initial display of retrieved documents than in a traditional, text-only display of ranked documents. The user can see which documents use the query terms in the same parts of a document, which documents use the query terms most frequently, and the relative length of documents. While TileBars doesn't attempt to change the documents that are retrieved from a query or the order in which the retrieved documents are displayed, the additional information it provides through visualization can help the user discover more quickly documents that are most likely to be relevant to his or her needs.

Other examples of systems that display one-dimensional data visually include Document Lens (Robertson & Mackinlay, 1993) (Figure 7), which maps multiple, reduced pages of text onto a three-dimensional shape, thus enabling a user to more quickly find and navigate to a particular page; and the Georgia Tech Information Mural project (Figure 8), which uses color, dots, and lines of varying magnitude to present a large amount of data in a compact space, while still providing access to the details of the data. All these systems show that even for simple, one-dimensional data, visualization can increase the usefulness of the information.

Figure 7. Document Lens
Source: [IEEE Computer Graphics and Applications]

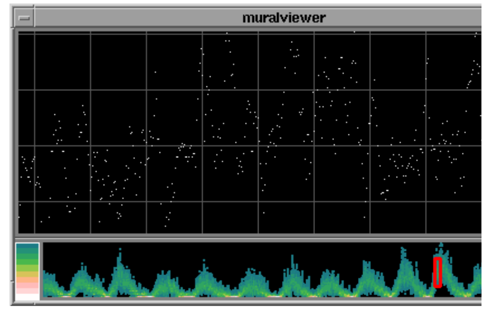


Figure 8. Information Mural
Source: [Georgia Tech Graphics, Visua
Usability Center]

Visualizing Two-Dimensional Data

In the context of information visualization, two-dimensional data is data that consists of two primary attributes that are represented in a space. Width and height represent the size of an item, for example, and the placement of an item on an x-axis and a y-axis represents a location in space. A geographic map with city locations, a floor plan of a building, and clusters of related documents in a document collection are all two-dimensional visualizations. (As described more completely in [Visualizing Multi-Dimensional Data], items in a two-dimensional data set can have more than two attributes. It is the fact that the data is described by two primary attributes, rather than the total number of attributes, that determines whether data is two-dimensional.)

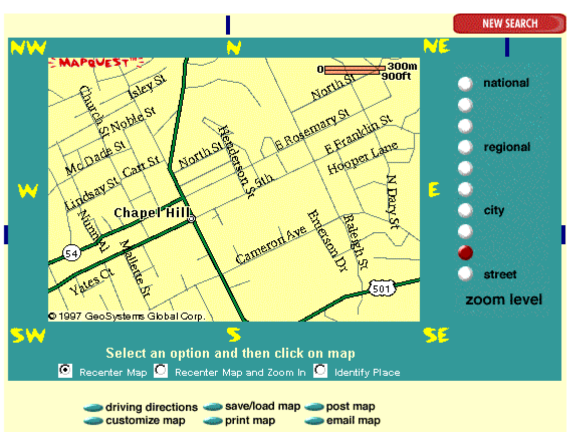


Figure 9. Geographic map
Source: [MapQuest]

The most common type of two-dimensional data visualization is geographic information systems (GIS). Large commercial GIS systems have long been used for regional planning, transportation planning and management, weather forecasting, and mapping. Simple GIS applications are becoming fairly common on the World Wide Web in the form of customized maps that are displayed to show the location of an address in response to a query to a search engine. Figure 9, for example, was returned by the [MapQuest] Web site in response to a query that supplied a zip code for Chapel Hill, NC and a street intersection. MapQuest provides 10 levels of zooming for displayed maps.

The traditional geographic map is an excellent example of how large amounts of data can be efficiently displayed visually. The fact that geographic maps were being drawn on clay tablets over 5,000 years ago attests to their power. Interestingly, despite this long history, it wasn't until the last half century that statistical data was plotted onto geographic maps. Once this technique was used, however, it became apparent how valuable a map so enhanced could be. During a cholera epidemic in London in 1854, Dr. John Snow plotted the location of deaths on a map, which revealed a concentration of deaths near a particular water pump. The water pump was found to be contaminated and removed from service, likely preventing more deaths (Tufte, 1997).

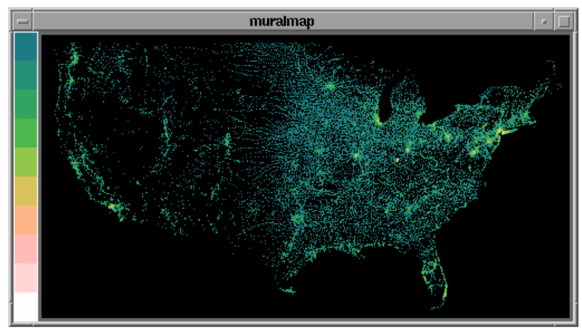


Figure 10. Visualizing population density
Source: Georgia Tech Graphics, Visualization and Usability Center

Plotting data, particularly health and census data, is commonly used today. A group at Georgia Tech uses a technique called the Information Mural to display data, such as population density (shown in Figure 10) accurately and automatically. In this example, dots on a map of the U.S. represent centers of population, with variations in color representing the density of population in each population center. As in the one-dimensional data examples discussed above, this type of visualization provides an overall view of complex data that enables the viewer to see patterns and connections that are difficult to discover in textual representations.

The geographic map metaphor has also been used in a more abstract way, resulting in interesting ways of visually information. For example, the Sitemap displays of Xia Lin use the map metaphor as a method of visualizing groups of documents in a collection, using size and location as the primary attributes (Lin, 1997). Documents in a collection are indexed and assigned to groups corresponding to the frequency of terms used within the documents and the collection. The relative importance of each group within the collection is represented in an abstract space (a large rectangle that serves as the map's outer boundary) by the size of the group's area within the bounding rectangle. In addition, the location of an area within the map indicates its relationship to other groups; neighboring areas are more closely related than groups farther apart from each other. Figure at 11 shows a Sitemap from a search on the Web using the query "UK and pensions."

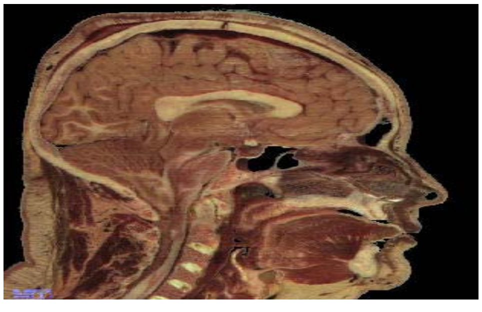
In addition to the size and location of the areas within the bounding rectangle, the size of the text within areas and the number and location of dots within the areas provide information about the document collection. These map displays have been used to represent different types of document collections, such as articles from conference proceedings, documents in a personal collection, and the results of search queries on the World Wide Web. In addition, sliders can be added to the map display, enabling a user to control the number of specific documents displayed as dots within their assigned groups, and to adjust the number of sub-group labels that are displayed in the primary group's area.

Visualizing Three-Dimensional Data

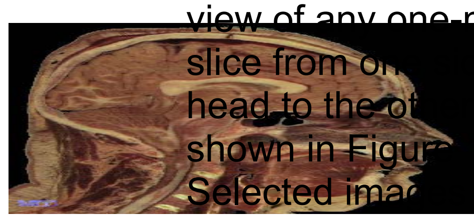
Three-dimensional data goes beyond two-dimensional data by incorporating volume. Many applications in scientific visualization are three-dimensional visualizations, because a primary purpose of scientific visualization is to represent real, three-dimensional objects. These computer models have provided scientists with a way to perform manipulations and experiments to predict how real-world objects actually behave, but which are too expensive, difficult, dangerous, or simply impossible to perform with real-world objects.

In recent years, three-dimensional visualization has been applied to a wider variety of areas, especially architectural and medical applications. The use of technologies such as QuickTime-VR, Virtual Reality Modeling Language (VRML), and digital imaging are used to create systems that realistically represent three-dimensional data. These systems enable people to examine and explore three-dimensional objects and spaces, in a way that is often more practical and efficient than going to the real thing.

For example, the National Library of Medicine's Visible Human Project provides a large digital library of anatomical images of the human body, based on precise, comprehensive images made from a male and a female cadaver at one-millimeter intervals. These images include color photographs of cross-sections and MRI and CT scans. A number of projects have built interactive, visual systems based on the [Visible Human Project] database.



The [Visible Human Head Browser] from the University of Michigan lets the user navigate through the images of the female head to see a



view of any one-millimeter slice from o e of the head to the ot as shown in Figur 2. Selected imag can be viewed at three resolutions.

Other projects based on the Visible Human Project data include [The Virtual Human](#) from the Argonne National Laboratory and the University of Chicago, (Figure 13), and the [Digital Anatomy Lab](#) from the Queensland University of Technology. The Digital Anatomy Lab has functions to show simultaneous two-dimensional views of anatomy parts from both the front and the side, and from both inside and outside of the skin, as well as three-dimensional animations (Figure 14 and Figure 15 ([MPEG movie](#) (1.1MB)). In addition to the projects discussed here, there are quite a few commercial applications available. Although these projects implement their visualizations of the Visible Human data in different ways, they are all using visualization to provide researchers and students a means to easily access information that would be very difficult to collect and understand working solely with books and slides.

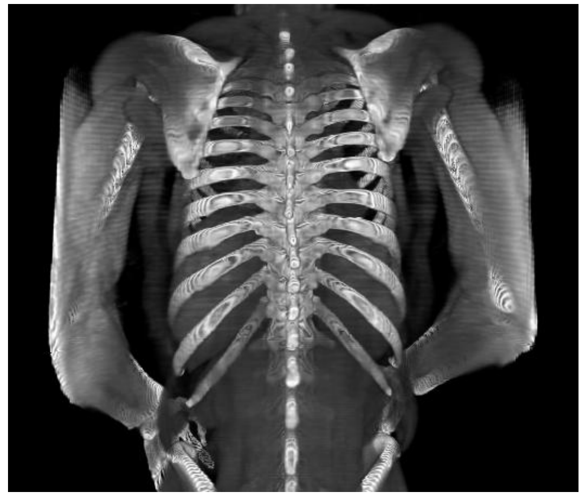


Figure 13. Three-dimensional skeleton
Source: [Argonne National Laboratory and the University of Chicago](#)

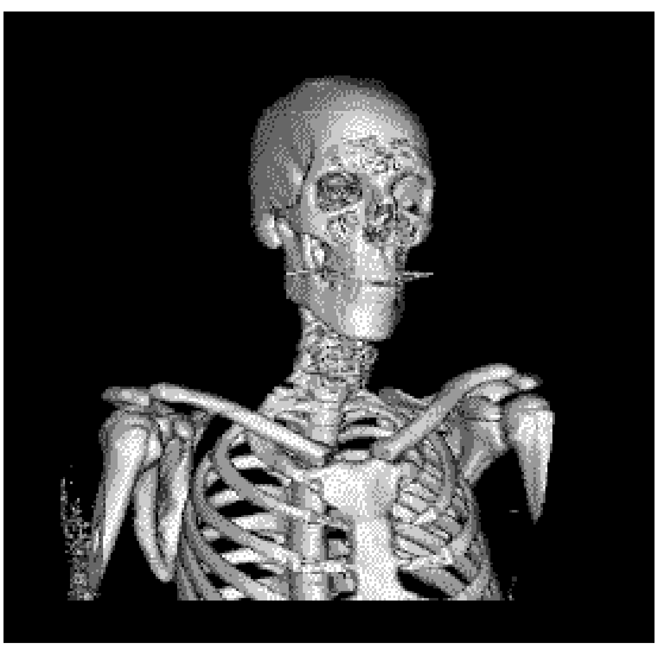


Figure 15. Movie of human hea
Source: [Queensland University Technology](#)

Figure 14. Animated skeleton
Source: Queensland University of Technology

The three-dimensional animation shown in Figure 14 is an example of how the increasing graphical and computational capabilities of computers not only enable more sophisticated versions of plotting data but give researchers a method to better explore and communicate large and complex systems. Thus three-dimensional and virtual reality models and animations are now being used to supplement or replace the maps, photographs, charts, and sketches that have traditionally been used to communicate this type of information.

Two projects that illustrate these possibilities use three-dimensional visualizations and simulations in the area of forest ecology. The SmartForest project uses an interactive modeling and visualization tool to simulate a three-dimensional forest environment. Starting with standard data – ground surveys, aerial photographs, and sample plot data – detailed three-dimensional visualizations are built, as shown in Figures 16 and 17. The SmartForest visualizations provide both high-level overviews of a forest and detailed views of individual trees at ground level. The representation of each tree in the visualization is a result of multiple variables in the database; changes to the database are reflected in the next refreshed visualization.

Figure 16. SmartForest forest visualization
Source: IMLAB, University of Illinois

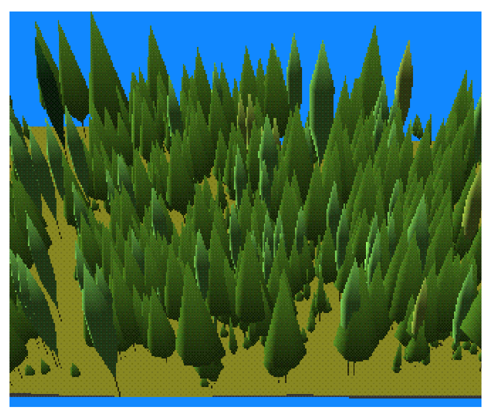


Figure 17. Forest visualization - tre
Source: IMLAB, University of Ill

The three-dimensional representation used in the SmartForest project has clear advantages over traditional GIS-based methods of showing such data. The three-dimensional representation of individual trees and the capability to view those trees at different scales provides a realistic view of the forest that can only be bettered by actually visiting the forest in person. At the same time, the ability to change the underlying database and immediately see the effects on the forest is something that can only be done in a computer simulation. This capability is particularly useful in complex systems like a forest, where many variables have interacting effects, but actually seeing the effects in a real forest can take decades.

Another project, SORTIE, is also based on a three-dimensional representation of forest ecology and provides a simulation of forest growth. SORTIE is oriented to showing changes to forests over time and is discussed in more detail in [Visualizing Temporal Data].

Figure 18. WebBook pages
Source: [Paper in Proceedings of CHI '96]

Other applications use three-dimensional data not to represent real objects but to provide a means of exploring and navigating information in a "natural" way, and in doing so enable users to access the information they want more easily. The WebBook and WebForager ([Card], Robertson, & York, 1996) applications use three-dimensional representations of documents and Web pages to improve the process of accessing information on the Web (Figure 18). WebBook collects related Web pages and displays them in a connected, three-dimensional view in which the user can see more than one page, flip through the pages, and directly access a desired page.

The developers of WebBook also created Web Forager. Web Forager is based on the same idea of presenting Web documents in a three-dimensional environment, but provides a workspace where the user can have multiple books available (on a desk or in a bookcase) for rapid access (Figure 19). Both WebBook and Web Forager use a three-dimensional representation of objects based on a familiar metaphor – a workspace consisting of books, a desk, and a bookcase – to provide the user with an

Figure 19. WebForager workspace
Source: [Paper in Proceedings of CHI '96]

easier, more intuitive way of accessing Web information. At the same time, these applications take advantage of the capabilities of computers to provide more ways of working with information than is possible in the real-world version of the workspace.

Visualizing Multi-Dimensional Data

Multi-dimensional data in the context of information visualization is data that describes an item with more than three attributes, each of which is more or less equal in the visualized context. For example, if you had a listing of the values of all the houses in the country and their address (one-dimensional data), you could sort and order the houses by value. You could also create a scale that showed the relative value of a house by the size of a dot representing the house and place the dots on a map of the country to show location (two-dimensional data). Although you may have information associated with other attributes of the houses, such as number of bedrooms, age, square footage, these are secondary attributes in the previous contexts and don't change the data to multi-dimensional data.

If, however, you have a database with a number of attributes of houses and the purpose of your application is to enable a user to sort the information about the houses using any of the attributes, it is multi-dimensional data. Researchers at the Human Computer Interaction Laboratory at the University of Maryland have developed a framework called dynamic queries (Ahlberg, Williamson, & Shneiderman, 1992) that enables users to visualize this sort of multi-dimensional data. The HomeFinder application, for instance, provides a visualization of multi-dimensional housing data.

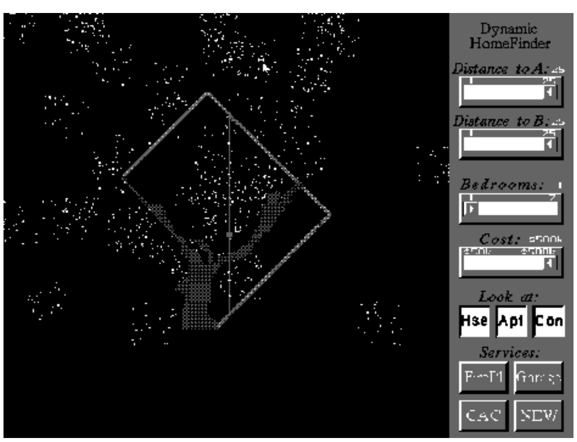


Figure 20. HomeFinder
Source: *University of Maryland, Human-Computer Interaction Lab*

HomeFinder displays the results of queries to a database of homes for sale in the Washington D.C. area. The user moves sliders that correspond to attributes in the database, such as price, number of bedrooms, square footage, and distance from a central location, as shown in Figure 20. As the user adjusts the sliders, the results of the query (dots representing houses that match the query) are dynamically updated on the displayed map. The user can select a dot from the map to display a detailed description of the selected house.

By providing a visualization of the database, applications such as the HomeFinder enable users to more easily see trends in the data and pinpoint exceptional data. Research from the University of Maryland shows that users performed tasks significantly faster using the visual, dynamic query interface as compared to ones where queries to the database were input through fill-in forms with textual output (Ahlberg et al., 1992). A number of subsequent applications have been developed with this framework of visualizing multi-dimensional data, such as films (Figure 21) and health statistics (Plaisant, 1993), and there is now a commercial product called Spotfire that provides users with more features and flexibility when working with their own multi-dimensional data.

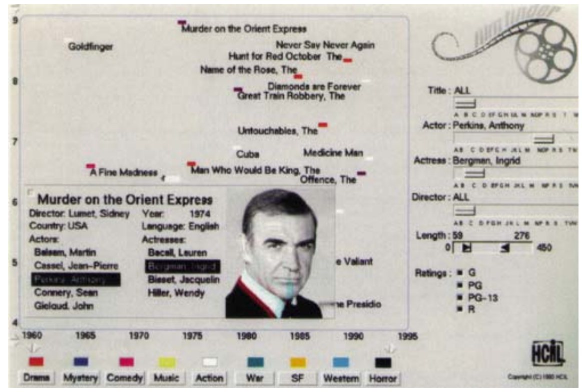


Figure 21. FilmFinder
Source: University of Maryland, Human-Computer Interaction Lab

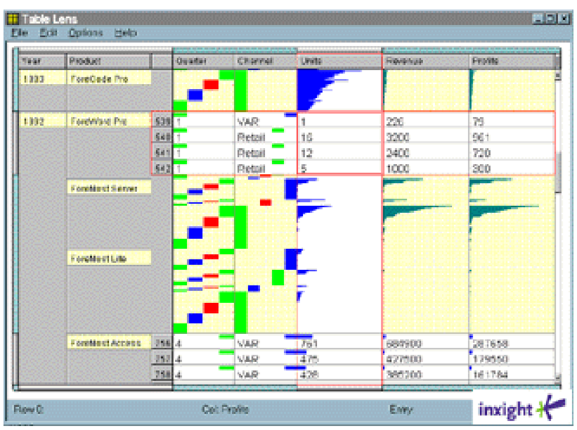


Figure 22. Table Lens
Source: inxight

The Table Lens, developed at Xerox PARC, offers another interesting way of visualizing multi-dimensional data. Table Lens combines the textual information contained in database tables with graphical representations of the same data, as shown in Figure 22. This allows the user to focus on specific information in a specific part of a table while still getting a sense of how that information relates to the larger context of the entire table. This scheme gives the user a way of looking at and navigating through the information that makes understanding the information easier (Rao & Card, 1994).

Visualizing Temporal Data

Graphical displays of data as it occurs over time is one of the most common and powerful methods of visualizing information and have been in continuous use for the past 200 years (Tufte, 1983). In recent years, the timeline as a basis for arranging data has become common in a variety of commercial software programs. Project management tools such as Microsoft Project use a timeline to enable the user to see at a glance the duration of events, when events occur in relationship to each other, and which events have dependencies on other events.

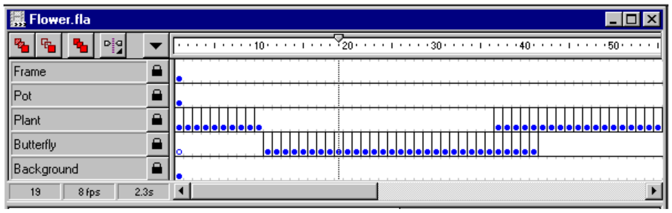


Figure 23. Multimedia timeline interface
Source: Macromedia

The timeline in multimedia authoring programs such as Macromedia's Director and Flash (Figure 23) provide a way for the user to synchronize one event to another so that, for example, music starts just as a title appears in the middle of the screen. These type of tasks are not impossible to do with a text-based interface, but the visualization provided by the timeline gives the user a much more intuitive way of working, and provides an overall view of events that is difficult to convey through text alone.

Another example of how powerful a visual timeline can be is illustrated by the LifeLines system developed at the University of Maryland. LifeLines provides an interface for visualizing biographical or personal history information. Figures 24 and 25 are examples from a medical application of LifeLines. In this application, the complete medical history of a patient is entered into a database. The LifeLines interface provides an overview of a patient's history on a timeline and provides tools for changing scale and focusing on specific details. Events, attributes, and relationships from the patient's entire available medical history are indicated by icons, horizontal lines, color, and line thickness.

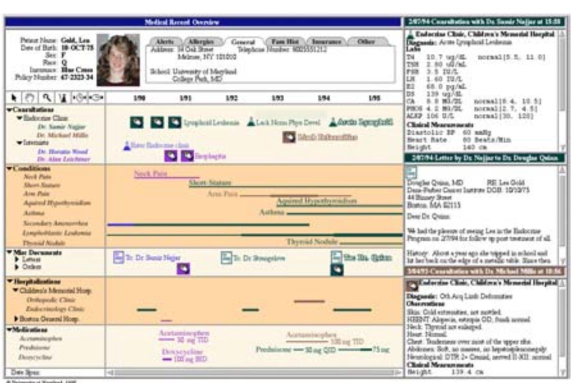


Figure 24. LifeLines
Source: University of Maryland, Human-Computer

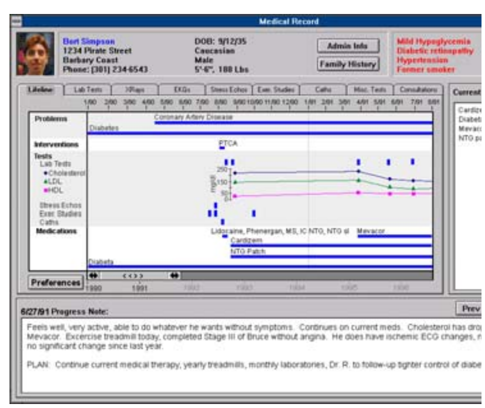


Figure 25. LifeLines - alternate int
Source: University of Maryland, Humai

[*Interaction Lab*](#) [*Interaction Lab*](#)

Experiment results suggest that overall, users are better able to comprehend and remember the information presented by the LifeLines visualization than with a tabular representation (Alonso, Rose, Plaisant, & Norman, 1997). The interface used by LifeLines is only one application of a framework that is designed to be applied to many types of personal histories, such as court records and professional histories, where the relationships between events are more complicated than can be represented by the simpler timelines used in project management or multimedia authoring tools (Plaisant & Shneiderman, 1997).

When temporal data is less abstract, it can be represented in more direct ways, such as with three-dimensional simulations. The SORTIE project is somewhat similar to the SmartForest project discussed in Visualizing Three-Dimensional Data. SORTIE is a three-dimensional model based on data collected in a forest in northwestern Connecticut. Unlike SmartForest, however, SORTIE doesn't use this data to precisely map an actual forest, but uses the data to create a realistic simulated forest, which can then be adjusted to examine forest dynamics over long periods of time.

The height, canopy width, species, and amount of light let through its canopy is represented for each tree in the forest. For example, Figure 26 shows an undisturbed forest at a particular point in time. This forest consists primarily of beech trees (purple) and hemlock trees (green), which provide a heavy canopy with low light availability. By adjusting parameters that control tree growth rate, reproduction rates, and light availability, researchers can simulate the effect disturbances will have on a forest over time. Figure 27 shows how the same forest develops after a disturbance, which enables yellow birch trees (yellow) to become much more dominant due to their higher growth rate and less dense canopies.

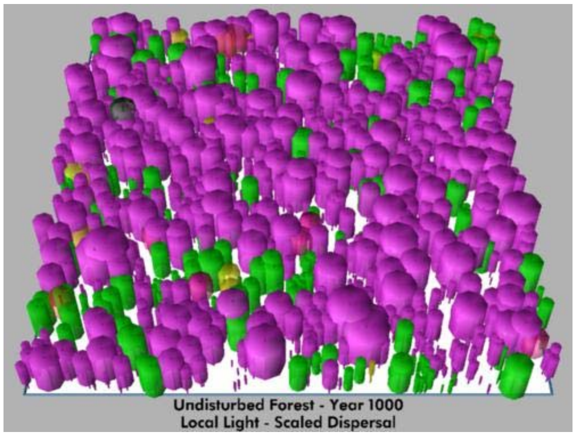


Figure 26. Undisturbed forest simulation
Source: SORTIE

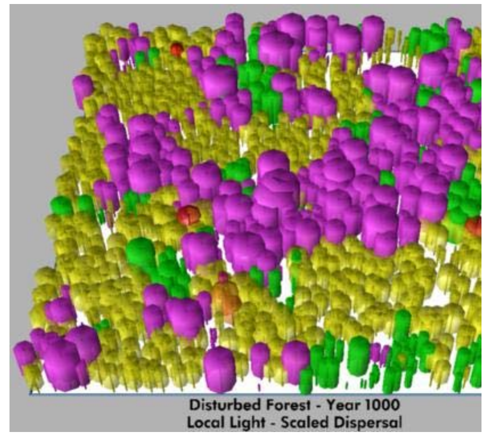


Figure 27. Disturbed forest simul
Source: SORTIE

Animations showing the changes in the forest over time can be very revealing, as shown by MPEG movies available from the SORTIE Web site (disturbed and undisturbed).

Visualizing Hierarchical Data

Hierarchical, or tree, data is data that has an inherent structure in which each item, or node, has a single parent node (except for the top-most or root node). Nodes can have sibling nodes (items that have the same parent node) and child nodes (items to which it is the parent node). Hierarchical structures are quite common. Business organizations, computer data storage systems, and genealogical trees are all examples of hierarchical data organized in a tree structure.

Basic visualizations of these structures are also common. The Windows Explorer interface to the Windows95 and Windows NT filesystems, for instance, displays the directory structure of a computer visually, enabling the user to more quickly understand the structure and navigate to a particular node than a text-based command-line interface. Tables of contents for Web-based documents increasingly use a visual display of a document's content to represent the parts of a document in a compact space. But these simple displays have significant limitations for many instances of hierarchical data.

For large hierarchical structures, the style of tree-view used by the Windows Explorer and other applications fails to represent the whole structure in a single view. When child nodes are collapsed into parent nodes in order to fit the whole structure into a single view, the child nodes become invisible; you don't know how many there are, where they are located, or what they are named. Expanding all the child nodes in a large hierarchy makes them visible, but requires the user to scroll through more than one computer screen to view all of the information.

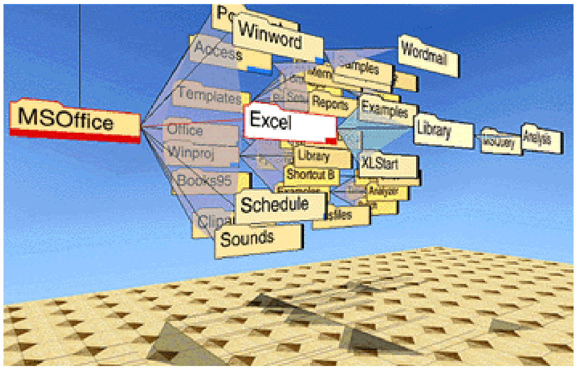


Figure 28. Cam Tree
Source: inxight

One approach to solving the problem of showing large hierarchies in a compact space is to represent them in a three-dimensional space, such as with Cone and Cam Trees developed by researchers at Xerox PARC. Cone and Cam Trees are hierarchies laid in three-dimensions, with the root node placed either at the top of the space (Cone Tree) or the left side of the space (Cam Tree). Child nodes are spaced equally in a cone shape extending below (or to the right) of the top node. Next level child nodes are likewise spaced equally in a cone shape beneath or to the right of their parent node. As shown in the Cam Tree example in Figure 28, labels of nodes are partly transparent to enable labels from nodes in the background to remain visible.

Cone and Cam Trees are not static hierarchical displays. When the user selects a particular node with the mouse, the selected node is highlighted and the tree structure rotates to bring it to the front of the view. This movement takes place slowly enough for the user to follow the movement, thus reducing the need for the user to reorient himself to the changed view. The entire Cone or Cam structure can also be rotated continuously, enabling a user to examine large hierarchical structures and understand the relationships within. Methods are provided to hide parts of the hierarchy and to move nodes within the hierarchy by dragging them to a new location. The amount of hierarchical information that can be represented by a single Cone or Cam tree is significant. The developers created a Cone Tree of a large organizational chart that showed on a single screen what took 80 pages to show on paper and nearly three feet of screen space to show with a conventional two-dimensional tree (Robertson, Mackinlay,

& Card, 1991).

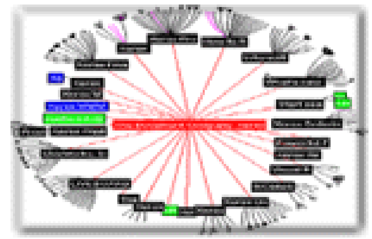
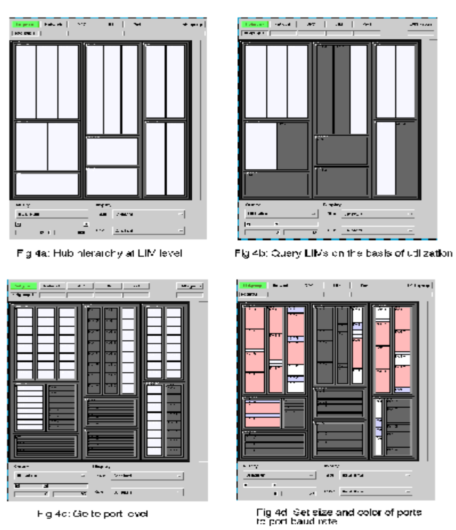


Figure 29. Hyperbolic tree browser
Source: inxight

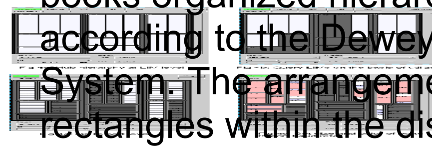
Even with a Cone or Cam Tree, it can still be difficult to represent very large hierarchies in a single view without obscuring some of the nodes. A variation of this approach, the Hyperbolic Tree Browser, deals with this problem by representing the entire hierarchy at a very small scale, ensuring that the user can see the entire structure, but magnifying the 10 to 30 nodes at the center of the screen, thus also enabling the user to see details (Figure 29). The display is dynamically updated as the user selects areas of interest within the hierarchy, thereby always keeping the part of the tree that the user is interested in magnified but in context with the rest of the hierarchy.

A problem faced even by three-dimensional displays of hierarchical data is that the size and content of each node is hidden; while you can see where in the hierarchy a particular node fits, all nodes appear to be equal. The treemaps approach to displaying hierarchical data attempts to solve this problem.

Treemaps are similar to the map displays of document collections described in Visualizing Two-Dimensional Data, except that those maps don't necessarily represent a hierarchical relationship. Tree maps represent the hierarchy of data through nested rectangles. All rectangles fit within a large bounding rectangle. Each of these interior rectangles represents a node; if it is a parent node it contains additional rectangles – child nodes – inside it. This design enables treemaps to show all nodes of a hierarchy in a single view. The number of nodes that can be shown by a treemap, in fact, is an order of magnitude more than in a traditional tree view (Turo & Johnson, 1992).



While representing all nodes of a hierarchy in a single view, treemaps are also able to display information about the individual nodes in the same view. The size of the rectangles in a treemap indicates their relative size within the entire hierarchy, and other attributes can be represented by color and a context-sensitive attribute display area (Figure 30). For example, a treemap has been used to represent a collection of



books organized hierarchically according to the Dewey Decimal System. The arrangement of rectangles within the display represents the hierarchical structure, the size of individual rectangles indicates the number of books at that level of the hierarchy, and the color within the rectangles indicates the frequency the books for that level have been used.

Visualizing Network Data

Network data refers to items (in some instances called nodes) that have relationships (links) to an arbitrary number of other items. Because nodes in network data sets are not restricted to a limited number of other nodes to which they link to (unlike hierarchical nodes, which have a unique parent node), there is no inherent hierarchical structure to network data, and there can be multiple paths between two nodes. Both items and the relationships between them can have a variable number of attributes.

Because the attributes and relationships between items can be very complex, network data is very difficult to show without some form of visualization. For instance, the Internet consists of tens of thousands of servers, with numerous possible paths between them. Attempting to understand Internet traffic patterns, peaks and valleys of usage, and alternate paths between nodes by looking strictly at tables and statistics, while possible for specific tasks, is made much easier by visual representations of this complex data.

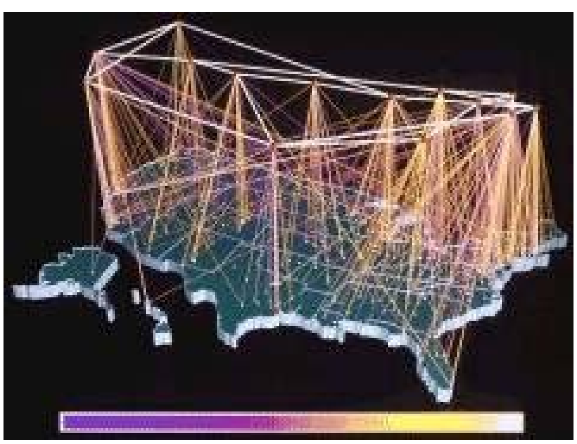


Figure 31. Network traffic visualization
Source: NCSA

Figure 31, for example, shows the volume of traffic on the NSFNET T1 backbone in September, 1991. Traffic volume is represented by colors ranging from purple (zero bytes) to white (100 billion bytes). By comparing visualizations like this for different time periods, researchers can easily see trends and patterns in traffic growth and detect areas of the network that show unusual rates of change.

There are a number of other on-going network visualization efforts, many of which focus on the Internet and related components. The work at Bell

Laboratories, as illustrated here in Figure 32, is a good example. Graphical network maps have also been used for many years in commercial network management products designed for enterprise networks. Networks managers use these graphical interfaces to monitor an enterprise's computer network and access software functions to troubleshoot and fix network problems. But network visualizations can also be used for types of network data that are completely unrelated to computer networks, such as grocery purchases (Figure 33), or the many interconnected components of software code.

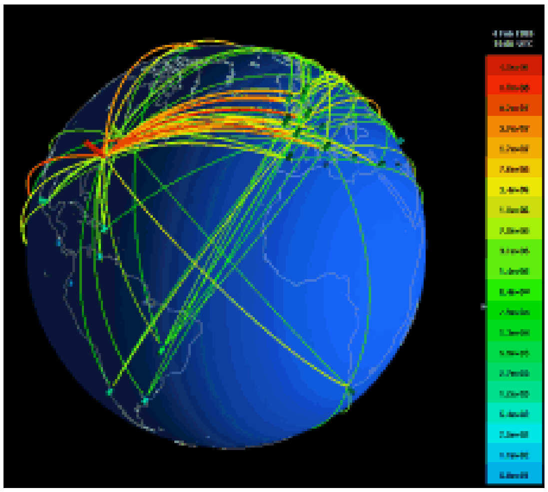


Figure 32. Global network visualization
Source: *Lucent Technologies*

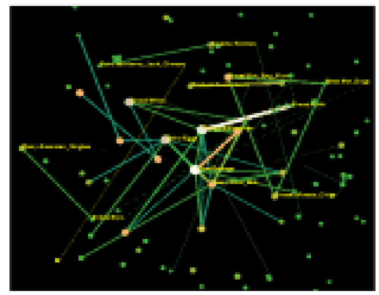


Figure 33. Associated grocery purchases visualization
Source: *Lucent Technologies*

The GraphVisualizer3D, developed at the University of New Brunswick, visualizes the complex relationship between the various components that make up a software program. In contrast to the one-dimensional representation of SeeSoft, GraphVisualizer3D uses network diagrams to illustrate how files, classes, variables, and functions (Figure 34) interrelate. By presenting a lot of complex relationships in a three-dimensional visualization, network diagrams of the type used by GraphVisualizer3D can help users understand relationships and dependencies in a large system more quickly and easily while reducing the need to look up such information in the code or in reference tables.

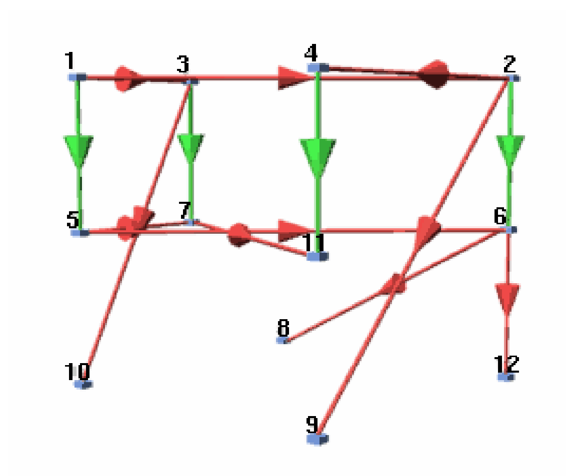


Figure 34. Visualizing software component relationships
Source: *University of New Brunswick*

The Future of Information Visualization

The projects described in this paper show that there are many ways information visualization can help people access information more effectively. Visualization systems enable people to look at large amounts of complex data and quickly find the information they want, to navigate and interact with data more easily, to recognize patterns and trends, and to obtain a better understanding of the information.

This doesn't mean that all these visualization systems will soon be in widespread use, however. Some visualization projects have been designed to explore a specific idea and are optimized to work with a very specific set of data. Making them work with more general data requires much more development work. Other systems, such as the SmartForest project, use extremely complex and detailed databases that are expensive and time-consuming to prepare (SmartForest, 1997). And although usability studies of some projects have shown that users are more effective when working with visualization interfaces, usability test subjects sometimes required a period of time to adjust to the visual paradigm; this might be a problem for systems aimed at the general public.

Still, a number of projects described in this paper have already evolved into commercial products. There are more than a dozen different visualization products available for the Visible Human Project data; the dynamic queries visualization research from the University of Maryland has been incorporated into a product called Spotfire; Xerox has formed a new company called inxsight to market products based on its ConeTree, TableLens, and Hyperbolic Tree visualization projects; and Lucent Technologies sells a variety of products related to its system of visualizing software code.

These commercial products indicate that there is a potential market for systems that incorporate visualization. And there are a number of trends in computing that will both encourage the development of information visualization applications and make these applications more likely to be accepted by computer users:

- Decreasing prices of multimedia-capable personal computers
- Increasing processing power of personal computers
- Increasing bandwidth and speed capabilities of computer networks
- Computer displays capable of displaying more complex and higher-resolution graphics
- Improved performance and functionality of fundamental software such as operating systems and Java

Equally important to the trends in technology is a growing awareness among information designers that for many forms of information, graphics can be "more precise and revealing" than traditional text-based presentations (Tufte, 1983). The need to deal with increasingly large amounts of information has prompted designers to begin incorporating more graphical information into their applications. The success of these applications, along with wider availability of tools for creating graphical information, is sure to lead to more widespread use of information visualization.

References

Ahlberg, C., Williamson, C., & Shneiderman, B. (1992). "Dynamic Queries for Information Exploration: An Implementation and Evaluation." In Proceedings of the ACM Human Factors in Computing Sytems (CHI '92), Monterey, CA.

Alonso, D. L., Rose, A., Plaisant, C., & Norman, K. L. (1997). "Viewing Personal History Records: A Comparsion of Tabular Format and Graphical Presentation Using LifeLines." Technical Report CS-TR-3795: University of Maryland Human Computer Interaction Laboratory.

Card, S. K., Robertson, G. G., & York, W. (1996). "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web." In Proceedings of the ACM Human Factors in Computing Sytems (CHI '96), 111.

Hearst, M. A. (1995). "TileBars: Visualization of Term Distribution Information in Full Text Information Access." In Proceedings of the ACM Human Factors in Computing Systems (CHI '95).

Lin, X. (1997). "Map Displays for Information Retrieval." Journal of the American Society for Information Science, 48(1), 40-54.

Plaisant, C. (1993). "Facilitating Data Exploration: Dynamic Queries on a Health Statistics Map." In Proceedings of the Annual Meeting of the American Statistical Association.

Plaisant, C., & Shneiderman, B. (1997). "An Information Architecture to Support the Visualization of Personal Histories." Technical Report CS-TR-3795: University of Maryland Human Computer Interaction Laboratory.

Rao, R., & Card, S. K. (1994). "The Table Lens: Merging Graphical and Symbolic Representations In an Interactive Focus + Context Visualization for Tabular Information." In Proceedings of ACM Human Factors in Computing Sytems (CHI '94), 318-322.

Robertson, G. G., & Mackinlay, J. D. (1993). "The Document Lens." In

Proceedings of the ACM Symposium on User Interface Software and Technology.

Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). "Cone Trees: Animated 3D Visualizations of Hierarchical Information." In Proceedings of the ACM Human Factors in Computing Systems (CHI '91).

Shneiderman, B. (1998). "Information Visualization." Designing the User Interface: Strategies for Effective Human-Computer Interaction (Third ed., pp. 522-541). Reading, MA: Addison Wesley Longman, Inc.

Small, D. (1996). "Navigating Large Bodies of Text." IBM Systems Journal, 35(3&4), 515-525.

SmartForest Project (1997). "http://www.imlab.uiuc.edu/smartforest/1.html." Imaging Systems Laboratory, Department of Landscape Architecture, University of Illinois at Urbana- Champaign.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information.* Cheshire, Connecticut: Graphics Press.

Tufte, E. R. (1990). *Envisioning Information.* Cheshire, Connecticut: Graphics Press.

Tufte, E. R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative.* Cheshire, Connecticut: Graphics Press.

Turo, D., & Johnson, B. (1992). "Improving the Visualization of Hierarchies with Treemaps: Design Issues and Experimentation." In Proceedings of the IEEE Visualization, Boston, MA.