Introduction

Information retrieval systems were originally developed from traditional retrieval systems such as the card catalog. In addition, most current systems were initially developed when text-based terminals were the standard. In practice, this is still the standard in many facilities, with graphic workstations only now replacing text terminals. As a result, most current information retrieval systems are text-based, using keywords to probe the database, and returning a list of hits on the database. These information retrieval systems suffer from a number of problems.

Korfhage (1991) provides the following list of potential pitfalls with traditional systems:

- They present to the user a limited set of documents, those judged most relevant. The judgment, however, is that of the system, not of the user.
- The retrieved documents are presented in a sequential list, often with little or no indication of the system's evaluation of each document. At best, the list is in decreasing order in terms of the single measure of relevance used by the system.
- There is no clear indication of the relationships that may exist among the retrieved documents.
- The user has little or no idea of the precise criteria for retrieval or nonretrieval of a document.
- What is presented to the user is often only a bibliographic record, not a full document.
- The user must assume that the system has made appropriate choices, and has little or no information about possible relevant documents that the system may have rejected.
- The user is treated as a uniform, faceless entity: there is little or no effort to take into account individual user differences.
- In many systems the user is left with little or no guidance in reformulating a query, should that be necessary.
- Even in systems with relevance feedback, the input that the user has in the feedback process is limited, and must be given with little guidance from the system. (p. 134)

Even when the user has retrieved a list of documents, it is difficult to tell just what has been obtained. Many questions can be asked about the list of documents returned. Has every relevant document been returned? What else is out there? What is the relationship of these documents to each other? What is the relationship of these documents to other documents not retrieved? If one of the documents returned is judged to be especially relevant, how can other related documents be identified?

Most current systems do not provide adequate answers to these questions. As information databases grow in size and user's expectations of information retrieval systems increase, our systems must do a better job of information retrieval. In addition, computing power is increasing at such a rate that strategies for information retrieval that would have been prohibitively expensive in terms of computing power and time are now becoming practical.

Many researchers are working on perfecting schemes that determine the user’s exact search requirements and that will return all documents and only those documents relevant to the user’s search request. Other researchers are working on schemes that will allow the searcher to easily deal with hundreds or thousands of documents at once, and to select relevant documents based on cues provided in the search interface. Graphical retrieval systems seek to increase the number of documents that can be displayed simultaneously and to allow the user to select relevant documents based on visual cues in the interface.

The Graphical Solution

In order to solve the problems inherent in current retrieval systems, Korfhage(1991) suggests that "the viewpoint [of information retrieval systems] should shift from retrieval to display" (p. 134). Fowler, Fowler, & Wilson (1991) write that "One of the common goals of visualization is to help manage and understand large amounts of data or information. From this perspective, information retrieval systems for large text databases are natural candidates for visualization techniques" (p. 142).

Instead of concentrating on returning just the set of documents that the user will deem most relevant, systems should concentrate on displaying the documents in such a way that the relationship between documents returned (and, if possible, those not returned) can be easily determined. The user can then use iterative refinement processes to home in on the set of most relevant documents. Instead of the system doing all the work, visual and graphical representations allow the user's own visual processing abilities to play a key role in the retrieval process.

Olsen, et. al (1993) describe some desirable features of graphical information retrieval systems:

- All documents should be presented via graphical representations in one display, thus avoiding the problem of forcing the user to create a holistic view from different displays.
- The position and other graphical features of a document's icon should intuitively give information on a document.
- Data reduction may be necessary, but the most important document attributes, as defined by the user should be retained in the display.
- The display should give an overview of the complete document collection retrieved, as seen from the user's perspective.
- Users should be able to identify single documents for retrieval of additional information.

- Users should be able to change the display interactively by relating document attributes to new graphical features, by viewing documents from a new perspective, etc. The transition from one perspective to the other should be apparent. (p. 73)

Shneiderman, with other authors, describes graphical retrieval systems based on the principles of direct manipulation, dynamic queries, and visual information seeking.

Direct manipulation systems are described as:

- Continuous visual representation of objects and actions of interest
- Physical actions of labeled button presses instead of complex syntax
- Rapid, incremental, reversible operations whose impact on the object of interest is immediately visible
- Layered or spiral approach to learning that permits usage with minimal knowledge. (Ahlberg, C., Williamson, C, & Shneiderman, B., 1992, p. 619)

An interface utilizing dynamic queries would:

- represent the query graphically,
- provide visible limits on the query range,
- provide a graphical representation of the database and the query result,
- give immediate feedback of the result during every query adjustment, and
- allow novice users to begin working with little training, but still provide expert users with powerful features. (Williamson, C., & Shneiderman, B., 1992, p. 339)

Visual information seeking designs contain:

- dynamic query filters: query parameters are rapidly adjusted with sliders, buttons, etc.
- starfield display: result sets are continuously available and support viewing of hundreds or thousands of items.
- tight coupling: query components are interrelated in ways that preserve display invariants and support progressive refinement. Specifically, outputs of queries can be easily used as input to produce other queries. (Ahlberg, C., & Shneiderman, B., 1994, p. 313-314)

Dynamic queries, starfield displays, and tight coupling will be discussed later in the sections that describe specific graphical interface solutions.

Specific Solutions

The following sections describe some experiments with graphical based displays of information. Some have been specifically designed to work as front-ends to information retrieval systems, while others have been designed to work in other information display tasks, but could be adapted to information retrieval systems.

I have divided the various information retrieval systems into several categories depending on the type of retrieval they support.

Retrieval for Hierarchical, User-Stored Data

Cone Trees

The Cone Tree, described by Robertson, Mackinlay, & Card (1991), is an information visualization technique in the Information Visualization framework (See Figure 1). It is another technique for visualizing hierarchical information. The hierarchical information is displayed in 3D in the shape of a cone, and either displayed vertically or horizontally. At the top of the cone is the item that is at the top of the hierarchical tree. Leading down from the top are items farther down in the tree. At each level down from the top, items are arranged in a circle below the item they are below in the tree. The cone shape is formed by lines connecting the parent node and the child nodes. The lines are filled in with translucent shading to make the cone shape more apparent.

In the vertical cone tree, the top node and its children form a shape like an umbrella. Items at the edge of the umbrella are arranged around the circle in three dimensions. Clicking on any item in the circle of items, causes the cone to rotate and bring the item in focus to the front. Any item with child nodes has a further cone radiating down from it. Clicking on any item rotates the whole cone tree so that the item of interest is brought to the front.

In this scheme, any hierarchical relationship can be displayed at once, with the cone tree able to handle up to 1,000 nodes with 10 layers. The viewer can also hide cones or relationships in order to make the relationships in other nodes stand out.

The Perspective Wall

The Perspective Wall is described by Mackinlay, Robertson, & Card (1991) as one of the techniques of information visualization that can be used in the Information Visualizer (See Figure 2). It is designed to be most useful in displaying information that can be laid out along some linear component. For example, files that have been stored on a hard disk can be displayed in the order of creation. Similarly, items that have been retrieved on the basis of one or two search terms and that can be described as a distance from the terms, can be placed on a linear structure. A second dimension can be added to differentiate between the items retrieved. In the case of files on a hard disk, the second dimension could be the type of file retrieved.

The Perspective Wall is displayed as a wall with three sides, joined at an angle of about 70 degrees. The middle section faces the viewer, with the other two sides angled away from the viewer in a 3D appearance.

In the display, an item can be selected as the focus of interest. The Wall is then rotated so that the item is at the center of the screen. Other related items are then displayed on the Wall at intervals from the focus, indicating their relationship to the focal point. Only the most closely related items are displayed on the center section of the Wall. If the viewer wishes to highlight another item, selecting that item causes the Wall to slide around so that the selected item is brought to the front and center.

Relationships between items in the system can be easily inferred, with the user retaining the power to focus on any one particular item.

The Information Visualizer

The Information Visualizer as described by Card, Robertson, & Mackinlay (1991) is a conglomeration of several different graphical display techniques. The immediate view is that of 3D Rooms. Each Room is a separate workspace with information relevant to a specific task enclosed in that Room. Within a Room, information display techniques such as the Perspective Wall, Cone Trees, or scientific data sculptures can be displayed, allowing further exploration of the information in a clearly prescribed context.

In order to work with the information in a specific Room, the user selects that Room with the mouse. Navigational aids are provided to allow the user to move freely from Room to Room. Color, lighting, shadow, transparency, hidden surface occlusion, continuous transformation, and motion cues provide different points of information about the data contained within a Room.

Card, S. K., Pirolli, P., & Mackinlay, J. D. (1994) have tested the Information Visualizer against traditional retrieval methods using a technique called the Cost-of-Knowledge Characteristic Function. This function is an attempt to work out a method to measure the information access in direct-walk interactive information visualizations. These types of systems provide for direct information access using only mouse selections and key selections.

A calendar based on the ideas embodied in the Information Visualizer was tested against a traditional UNIX calendar. In this experiment, the unit of cost being considered was primarily the user’s time. The question put to the user was to view the data from a specific day, where the day to be viewed was at different lengths of time from the current day. The cost to use the Information Visualizer calendar increased arithmetically, while the cost to use the traditional calendar increased geometrically. Thus, the traditional calendar took a shorter time to access the day when the day to be accessed was relatively close, but the Information Visualizer calendar took less time when the day to access was very far from the current day. It appears that utilizing techniques embodied in the Information Visualizer should make it relatively easier to explore a large document space than traditional techniques.

The Information Grid

The Information Grid (Rao, et. al, 1992) describes a system for searching a document space where images of the documents are stored. The system allows the user to enter search terms in a query area. Retrieved documents are then shown in a grid of fixed-size thumbnail sketches, where thumbnails are generated from the first page of each document. Thumbnail documents can then be used to seed further searches by selecting suitable documents from the results area, or from a holding area where documents have been dragged for storage.

Protofoil

Protofoil (Rao, et. al, 1994) builds on the work described in the Information Grid to handle paper documents (See Figure 3). It stores, retrieves, and manipulates paper documents as electronic images. The four components of the system include:

- a document database for storing documents as page images along with other information needed to support search, retrieval, and browsing of the documents.
- document services for printing, emailing, faxing, and OCR.
- scanstation user interface for storing documents into the document database and for otherwise accessing documents or invoking services using paper user interfaces.
- workstation user interface for retrieving and browsing documents and auxiliary information and invoking services. (Rao, Et. al, 1994, p. 182)

The Protofoil system provides for document retrieval “based on an iterative loop in which four user actions are variously interleaved: selecting a scope (initially the entire file cabinet); specifying a query; browsing results; and browsing and using documents.” (p. 182) Searches return results in different five views:

- Thumbnails - array of small pictures of the documents returned.
- Description - a list showing attributes of the documents returned.
- Category Groups - grouping of documents according to their category.
- Clusters - list of automatically generated clusters of documents, with each cluster showing the number of documents in the cluster, the central terms of the cluster, and a few central documents from the cluster.
- Snippets - list of documents showing snippets of the document with search terms in the document highlighted in the view.

Results can be fed back into the search query to further refine the search. As with the

Information Grid, this is accomplished by selecting relevant documents, then marking them to be used as a seed for future searches. Once documents are located, they can be placed in a holding area for future use.

MUE

MUE (Museum Unit Editor) (Travers, M., 1989) is a front-end to the Cyc database at MIT (See Figure 4). The Cyc project is “an effort to build a very large knowledge base that encompasses a broad range of common-sense knowledge. The knowledge base consists of a network of interrelated units (frames), each of which corresponds to a thing to be represented. These can be physical objects, abstract concepts, classes, or anything else of interest.” (p. 147)

Information in the database is organized as an unrestricted labeled graph, but parts of it are defined so as to form directed acyclic graphs. MUE uses these to provide navigational skeletons. The directed graphs have relationships of more general to specific and more specific to general. The general to specific relationships can be arranged in the form of a tree where every arrow points downwards.

The metaphor chosen to display this data was that of rooms in a museum. Every unit in the database is mapped as a room in the museum space. In order to display an individual unit, the tree must be re-rooted to place the selected unit at the top of the tree. Arrows that point up in the new arrangement can be replaced by negative arrows that point downward to form a new tree relationship. This can now be displayed as a series of enclosed spaces, with each level in the tree at the same level of enclosing room. Rooms that point from general to specific can be displayed as a different color from those that point from specific to general to show the relationship between units of knowledge. Because the knowledge contained in the database is organized as a network, in some sense, all items in the database can contain every other item in the database, so every item in the database can be displayed as a room that eventually leads to every other room.

Navigation through the database consists of following leads and relationships shown by the rooms in the “museum.” To follow a lead, simply open a room and move to another level. All movement through the database is through hypertext-like navigation.

Fisheye Views Into Data

Generalized Fisheye Views

One of the earliest description of a fisheye view for displaying relationships between items of information was given by Furnas (1986). In his paper, Furnas describes a technique for focusing on items of direct interest while retaining some view of items of less interest. The most information is given about the item in the focus. Less information is given about items farther from the focus, while very little information is given about items far from the focus. If the focus is shifted to another item, then the amount of information given about different items changes to reflect the new focus.

Graphical Fisheye Views of Graphs

While Furnas provided only text-based examples, Sarkar & Brown (1992) and Brown, Meehan, & Sarkar (1993) provided graphical examples of fisheye views (See Figure 5). In their work, an information retrieval task was completed and items of information returned. The information was then displayed as a network picture, with lines connecting related items. The item with the focus would be displayed as the largest item in the display, with the size of items decreasing the further they were from the focus. Other characteristics of relationship could be displayed by distance from the focus, so that both distance and size could indicate strength of relationship.

Magic Lens Filters

The Magic Lens filter (Stone, M. C., Fishkin, K., & Bier, E. A., 1994) is a tool to view an underlying model of visible data. Like the generalized fisheye view, the data under the lens is distorted to make relationships more apparent. The data can be in a highly compressed visible format, or can be completely different from the view of the data as viewed through the lens. As the lens is passed over the data, an algorithm specific to the lens is applied to the underlying data for the visible model and the modified data is displayed in the lens portion of the interface.

Similar to a real magnifying glass, the Magic Lens can magnify a portion of a picture or text. As the lens is passed over a map, for instance, the portion of the map under the lens can be magnified to show more detail. If the visible area is text, it can be magnified to be more easily readable.

In other applications, the lens might be set to show only certain words or match certain variables in entries that the user wants to see. As the lens is passed over the data, only those entries in the data that contain the word or match the variables will be made visible. All other entries will remain in the background. For example, when viewing a map, the user could set the lens to highlight all schools. As the lens is passed over the map, only those buildings that are marked as schools would be highlighted in the lens.

In a search retrieval application, all the entries in the search space could be visible on the screen, for example as colors or points in the search space. The lens could then be set to find entries based on some search criteria. As the lens is passed over the search space, only those entries matching the search criteria would be highlighted.. The user could then call for more information about those specific entries in order to explore further.

The Table Lens

The Table Lens (Rao, R., & Card, S. K., 1994) is used for visualizing tabular data. As a fisheye technique, it involves distorting parts of the table to make the underlying data in the table more visible. As with other techniques described here, the Table Lens fuses graphics and text in order to increase the amount of information that can be displayed at one time in the table. Compared to a normal table, the Table Lens can display up to 100 times the information on a single screen.

In practice, a table is constructed containing a number of rows and columns. The table might be sorted on a specific column in order to highlight one facet of the data. The data is then viewed graphically, so that values in each cell in a column are replaced by a line representing the value in the cell. Relationships can be discerned at a glance by looking at the chart. A specific row or column can then be expanded if the user wants to explore the data in that row or column in more detail. At this point, the text view of the data can be viewed in the row or column to see exactly what the data point says. This way, the user can explore the data, zeroing in on specific points to see exactly what underlies the visible relationships.

Value Bars

Value Bars (Chimera, R., 1992) are most useful in displaying information about items in lists and tables that have multiple attributes. It is possible to sort the information in the lists or tables, but the list or table can be sorted on only one attribute at a time. Value bars placed next to the data can provide information about each attribute in the list or table, without having to sort each one separately.

As an example, a disk's directory can be displayed with values bars. Files can have information relating to size and days since creation. The files in the display can be sorted by name, and on the side of the file display two value bars can be placed showing file size and days since creation. The value bars display cells corresponding to the values for file size and days since creation. In addition, the value bars show data for all entries in the list, even those that are not currently in view in the text view of the files sorted alphabetically. Each bar is also marked so that the cells corresponding to the visible files is shown. The user can get an immediate overview about file sizes and days of creation of the files. In addition, the user can click on a cell in the value bars to go directly to a file that has a size or creation data that looks interesting.

Treemaps

Generalized Treemaps

Johnson & Shneiderman (1991) describe a method of displaying hierarchical information in fullscreen displays called treemaps (See Figure 6). In a later paper Johnson (1992) describes a specific implementation of treemaps called TreeViz. Treemaps are designed to display hierarchical, tree information in a graphical format. The appearance of a treemap is similar to a

Venn diagram. Items farther down in the tree are enclosed within walls representing items farther up the tree. An item at the bottom of the tree (a leaf node) is shown in a size representing either the size or importance of the item.

When applied to a structure such as a directory of a hard drive, the picture presented shows the sizes of the files and the relationship between the subdirectories and the files in the directory. If the treemap were applied to a hierarchical structure such as the Library of Congress Subject Headings, the relationships between subject terms could be visually displayed.

Turo (1994) describes a use of treemaps to display information about the NBA. All information for all players, teams, and the four divisions of the NBA are entered in a database. A treemap is then generated to display the information, sliced by division, team, then players. The user can then emphasize different aspects of information about the players, such as points scored, height, years played, in order to make relationships clearer and to make certain players stand out. As each player attribute is emphasized, fisheye techniques are used to make the emphasized player block larger so that it can be seen and selected more easily.

TreeViz

Johnson (1992) describes a treemap applied to a hard drive containing more than 1000 files. All the files and the relationships between the files can be displayed on a single screen.

Starfields and Dynamaps

Dynamic Queries

Dynamic queries are contrasted with text search queries into a database returning a text listing. Dynamic queries (Ahlberg, C., Williamson, C, & Shneiderman, B., 1992), embody the principles of direct manipulation and visible actions as listed earlier. Dynamic queries include the use of graphical widgets, such as sliders, to enter a query and graphical output, such as a map, to display the results of the query.

Ahlberg, Williamson, & Shneiderman describe a dynamic query system based on the periodic table. The periodic table is displayed on the screen with sliders controlling search probes into the table. Values can be changed for atomic mass, atomic number, atomic radius, ionic radius, ionization energy, and electronegativity. As the sliders are adjusted, the value corresponding to the slider's position are displayed. At the same time, elements in the table are highlighted if they satisfy the criteria as set by the sliders.

Selection of values is immediate, and changes are visible immediately. The user can change the values and see the results of any change immediately. In addition, the graphical widgets permit the user to see what the maximum and minimum values are that can be affected by the widget, so that the user does not have to guess at permissible values.

Current sliders usually deal only with numbers. In order to deal with ordered text, the text is usually placed in a scrolling list. The user then scrolls through the list and selects the desired entry from the list in order to use it elsewhere. Ahlberg, C., & Shneiderman, B. (1994a) describe the Alphaslider. The Alphaslider is used to display ordered alphanumeric data in a slider, in the same way that numeric values can be displayed. Letters are placed below the slider, with spacing between the letters indicating the number of entries under each letter. Text output is displayed above the slider. As the user moves the slider thumb, the text above the slider changes as text values are accessed through the slider. This allows text as well as numeric values to be used in dynamic query systems.

Dynamaps

Dynamaps (Plaisant, C., & Jain, V., 1994) are implementations of dynamic queries applied specifically to maps. A dynamic query system is constructed with a map of a region as the display area. As the sliders representing search probes are manipulated, areas on the map change in response to the values that are returned. Different gradations of colors are continuously applied to the map areas to represent the values selected. The user can move the slider thumbs and instantly determine how changing values change their effect on the map.

Starfield Displays

Williamson, C., & Shneiderman, B. (1992) introduce what they would later term a Starfield display with the Dynamic HomeFinder. Starfield displays join two important concepts:

- dynamic queries
- tight coupling
- all values in the search space visible on the screen at the same time

Starfield displays are constructed so that all points in the search space are shown as points in the display field. By selecting values to search on, the user adjusts the display so that more or less points are shown on the screen.

The Dynamic HomeFinder illustrates the concept with an application that allows a user to select the house of their choice. A base map of the Washington, D. C. area is presented on the screen. All available houses in the area are presented on the map as single points. Sliders and buttons allow the user to adjust distance from selected points on the map, number of bedrooms, cost, type of residence, and services provided. As the user adjusts the sliders or selects buttons, the map is adjusted to show only those houses that satisfy the selected criteria. As the values are adjusted, the map immediately reflects the selected values, allowing the user to explore houses on the map with minimal cost. If the user finds a point of interest, the user can select the point to get more information about the house.

In a later article, Ahlberg, C., & Shneiderman, B. (1994b) describe a further application of the Starfield display. In addition to the Dynamic HomeFinder, also described is the FilmFinder (See Figure 7). The FilmFinder displays all known films in the search space as points in a two-dimensional grid with year on the x-axis and popularity (scale of 0 to 9) on the y-axis. Alphasliders are used to adjust the scale for such variables as title, actor, and actress. Normal sliders adjust length of film. As the number of points on the screen is reduced by manipulation of the sliders, there comes a point where there are few enough points that the information associated with each point can be expanded. At this level, there is enough free space on the screen to place the name of each film beside its point on the screen. If the user wants more information about a film, the user can choose the film and have a dialog pop up with detailed information about the film. The user is allowed to choose values on this dialog to feed back into the search probe to further define the points that show on the screen.

Graphical Views Into Large Databases

Pathfinder Networks

Fowler, Fowler, & Wilson (1991) incorporate some ideas from fisheye views in their information browser (See Figure 8). Their system is specifically designed to display the network of relationships that can be formed when the searcher enters a natural language query. Once the relationship diagram is formed, the searcher can select one of the nodes and bring it to the center. Related nodes are then displayed, in ellipses, surrounding the focus. Nodes that are farther from the focus and weaker in relation to the focus, will be displayed as smaller ellipses.

Articles closely matching the focus node are displayed in a separate viewer. The searcher can choose to move the focus in the network relationship, bringing different sets of articles to the fore, or the searcher can choose one of the article titles for an overview of the article or for retrieval. The system view can be switched so that article titles are displayed in the network relationship. Then, related article titles are displayed, with network lines and the size of the article icons indicating their relationship to the focus node. Again, the focus can be shifted to other articles, and new, related articles will then be brought into view.

GUIDO

The GUIDO system, described by Korfhage (1991), is designed to work with reference points that are numerical values assigned to documents. For example, a document could be assigned a value based on the number of times a search term occurs in the target document. The searcher defines several points to use for the search. The system then returns those documents that show a value for any of the points. If two points are used, then a flat document space is defined, with the documents placed in the document space depending on their numerical relationship to the reference points. The higher the numerical value, the closer the document to the reference points.

If three points are used, then a three-dimensional document space will be formed. N reference points will produce a n-dimensional document space. Since more than three dimensions are difficult to display, colors, shape, size, movement, and sound are required to represent more dimensions. The position of the documents in the document space show the relationship between the documents returned.

The user can view the document space and determine the level of relationship that appears to be useful. Once a level of relationship is determined, the searcher can set a cap on the document space and obtain all the documents enclosed in the newly capped document space. Relationships are easy to determine using GUIDO as long as the number of reference points does not grow too large.

VIBE

Korfhage (1991) and Olsen, et. al (1993) describe a two-dimensional document space called VIBE, that is designed to handle a larger number of reference points than is feasible with GUIDO (See Figure 9). Distances between reference points are calculated as ratios rather than as absolute values. Points representing documents are then projected onto a two-dimensional space. Documents that are closely related will form clusters in the document space. Documents that are strongly related to a particular reference point will be closer to that point.

Once the documents are placed in the document space, the searcher has complete freedom to reposition the reference points in the diagram. This might be necessary in order to clarify relationships between documents and between documents and different reference points. Since all the reference points are placed in a two-dimensional space, GUIDO'S problem of displaying more than three dimensions is avoided.

More information about each document in the document space can be obtained by clicking on the document icon. Once an appropriate document or cluster of documents has been selected, the full text of the documents can be retrieved.

Bead

Chalmers & Chitson (1992) describe a three-dimensional information retrieval viewer called Bead (See Figure 10). A large number of variables can be used to describe documents in a database - either determined manually or automatically. Latent semantic analysis and multi-dimensional scaling are then used to set up rules of physical behavior for each document. In effect, each document becomes a particle in the document space with force and motion relationships produced between particles.

When the user enters a search request, a set of documents is returned, their positions determined by the rules calculated earlier. The documents are represented as a cloud of numbered points in a three-dimensional document space. Documents that are most closely related to the search request are highlighted in a sphere of interest, while other documents are dim. Documents will tend to form clusters in the cloud of documents in the document space.

The user can zoom in on any particle in the document space in order to see neighboring particles and how they relate to each other. When a particle becomes the focus, a new sphere of interest is formed and a new set of particles is highlighted. The user can move through the document space, investigating relationships among the particles in the space.

In addition, it may be possible to rotate the document space to uncover other relationships. Particles that appear to be closely related on one dimension in the three-dimensional space, could be revealed to be far apart on other dimensions when the document space is rotated.

The searcher can select any particle, in order to investigate information about the document represented by that particle. Once a set of particles has been determined to be of interest, the documents represented by those particles can be requested by the user.

InfoCrystal

InfoCrystal (Spoerri, A., 1993) is described as a visualization tool that can also be used as a visual query language to help users search for information (See Figure 11). InfoCrystal has the following properties:

1) Users can explore an in formation space along several dimensions simultaneously without having to abandon their sense of overview.
2) Users can manipulate the information by creating useful abstractions.
3) Similar to a spreadsheet, users can ask “what-if” questions and observe the effects without having to change the framework of a query.
4) Users receive support in the search process because they receive dynamic visual feedback on how to proceed. They can selectively emphasize the qualitative or the quantitative information provided by the feedback to help them decide how to proceed.
5) Users can formulate queries graphically, and they have flexibility in terms of the particular methods use to retrieve the information. For example, users can seamlessly move between a Boolean and a vector-space retrieval approach, or they can easily switch from a keyword-based to a full-text retrieval approach. (p. 151)

InfoCrystal is based on the concept of Venn diagrams, except that it is easy to extend the visual picture to more than three variables. As an example, the following picture of a three variable InfoCrystal is described:

- Construct a Venn diagram with three circles.
- Divide the Venn diagram into seven pieces based on the areas delineated by the intersection of the circles.

- Convert the seven pieces into icons based on the number of variables intersecting to produce the piece. For example, the portion of the Venn diagram representing one variable only (no intersecting pieces) is represented by a circle. The portion representing the intersection of two variables is represented by a rectangle, and the portion representing the intersection of three variables is represented by a triangle. For a larger InfoCrystal, the number of sides of an icon represent the number of variables intersecting.
- Place the icons in an enclosing shape representing the number of variables (triangle for three, square for four, etc.) in a position that indicates which variables are being represented.

Shape, Proximity to variable, Rank (distance from center), Color, Orientation, and Size or Brightness of icons are all used to relay information about the variables they represent. Since the icons in the InfoCrystal represent the results of various Boolean queries, the users can activate the queries by simply clicking on the icons. The user can activate the query represented by the icon, or by choosing several icons at once, can activate a complex query as the intersection of the individual queries.

In addition to the basic features described above, users can modify the queries in the following ways:

- Hierarchical structures can be built where multiple InfoCrystals interact. The output of one InfoCrystal can be used as the input to another InfoCrystal.
- Retrieved documents can be used as an input criterion.
- Users can modify the search result with the use of sliders that represent relevance weights and selection thresholds. By adjusting the sliders, the user can determine which icons are selected.
- The InfoCrystal layout can be changed to a “Bulls-Eye” layout that places relationships with a higher relevance score closer to the center of the InfoCrystal.
- The InfoCrystal layout can be shown in the “Bulls-Eye” layout with all documents rather than just the relationship icons.

LyberWorld

LyberWorld (Hemmje, M., Kunkel, C., & Willett, A., 1994) has been designed to make use of 3-D visualizations into a database of fulltext documents (See Figure 12). It combines ideas from Cone Trees and the Vibe system, and extends the Vibe system into a 3-D Relevance Sphere for display of relationships.

The contents of the documents in the system are analyzed and the documents are placed into a hypertext relationship based on the analysis. Each document can have links to many other documents in the system, creating a tangled web of interrelationships. In use, when a search term is entered to start the search process, the web is converted to a tree, in a similar manner to that employed by MUE.

When a user enters a search term, that term is made the root of the tree, and all documents and search terms directly related to that term are then transformed so that they appear as child nodes off the root. All closely related nodes are then similarly transformed and placed into the tree. The rest of the web could be similarly transformed and placed into the tree, but only a small proportion of the entire web is necessary for initial display. Once the tree is formed, the first few levels are displayed in a cone tree. The use can then browse the cone tree looking for interesting titles or interesting pathways down the cone tree, following a related search term down a level in the tree. As the user moves towards unexplored regions, the cone tree can be unfolded to show more pathways (at which point, the calculations are done to transform the web into more pathways in the tree). If the user finds an unfolded path uninteresting, the path can be reduced again. If the user finds a title of interest, the document can be opened to determine if it really meets the needs of the user.

After the user has explored some levels of the cone tree content space, and has unfolded areas that are interesting, the user can switch to the Relevance Sphere. The term nodes selected, with their related nodes, and documents relating to the term nodes are placed into a sphere with the term nodes on the surface of the sphere and the related documents inside the sphere. Relevance relationships for the term nodes are calculated as in the Vibe system. However, where Vibe displayed the terms and documents in a 2-D space, the Relevance Sphere introduces a third dimension in order to clarify relationships between documents and term nodes. The more strongly related a document is to the term nodes, the closer it will be to the surface of the sphere. Three types of node/sphere manipulations can be done in order to investigate document relatedness.

- Term nodes can be moved around on the sphere in order to see which documents move with the nodes and to investigate the interaction between term nodes.
- The sphere can be rotated in order to separate clusters of documents.
- The “attraction” of term nodes can be increased or decreased (as with Treemaps), to separate documents based on how strongly they are attracted to that particular term node.

At any point in the manipulation of the nodes and documents, the user can switch views from the cone tree, to the Relevance Sphere, to a view of the document itself. In whatever view, the context of the search is preserved and transferred into the other tool’s visualization metaphor. The user is free to conduct the search using whichever tool is the most comfortable, and which will yield the best results at the time.

The VisDB System

As with Starfield displays, the VisDB system (Keim, D. A., Kriegel, H, & Seidl, T., 1993) attempts to place all points in the database on the screen at the same time (See Figure 13). The authors recognize that there is an upper limit on the number of points that can be displayed, relating to the number of pixels in the display portion of the screen. For search results consisting of more points than can be displayed, each pixel on the screen would represent more than one point.

To start the system, the user specifies a single query. Once the user gets a result, the display can be modified with other query terms or by manipulating the range of the query.

The user enters a search query, returning a set of data as a result. The relevance of each item to the search term is then calculated and a value assigned to the item. A display point is then generated for each item returned and the points displayed on the screen depending on their relevance value. Those items with a greater relevance are placed nearer to the center of the screen. In addition, color is used to differentiate between groups of relevance values, with the most relevant items colored yellow, with shading to green, blue, red, then to black as least relevant. By looking at the display, the user can immediately determine the proportions of documents returned by relevance to the search term.

A specific color is related to a specific range of relevant values. These values are displayed on sliders, which correspond to each of the search terms. The user can vary the range of values covered by each color, thereby modifying the search result and the corresponding display. The user can thus gain immediate feedback for additional search probes into the returned data set. Different kinds of sliders can be set up for numbers, ordinal values, non-numeric data, or for special data types. To focus more closely on one area in the display, the user can select a specific color and enlarge its area. Once an interesting set of items is found, the user can view the specific items selected.

Discussion

Standard for Graphical Information Retrieval Systems

How well do the systems described here conform to the standards for graphical information retrieval systems as described by Olsen? (see Section 2 above)

All the systems present their data graphically, and for reasonable amounts of data, all the data is displayed in a single screen.

Positioning of a document's icon in the display of every system gives some information about the relationship of that document to other documents in the system. In some cases, as with Cone Trees, the relationship is hierarchical. In other cases, as with Bead, InfoCrystal, or LyberWorld, it shows how far apart the documents are on some characteristic defined by the system.

Data reduction is used by every system. In all cases, this serves both to fit the information onto a single screen and to reduce the amount of information that the user must absorb when trying to make sense of the relationships displayed. However, all systems provide a mechanism for the user to display more information about a single item in the system. In many cases, selecting the item pops up a little window with more information about the item. In some cases, selecting the item allows for the actual content to be displayed in a window on the screen. It does not appear that it is possible to create a system that will display all important document attributes while simultaneously displaying hundreds of data points on a single screen, without making the display impossible to read.

All systems provide for the entire retrieved collection to be displayed on a single screen, along with the relationships between the documents, except for the MUE and LyberWorld. However, in both cases, the entire collection is available for the user to jump to via hypertext links. Navigation feathers make the entire system available, but only a present a small portion in order to make the system more manageable. As with the other systems, many more points are displayed than in conventional systems.

All systems provide the capability to highlight a single item and retrieve and display more information about the item and if required, to retrieve and display the item itself.

All systems provide the capability to change the display interactively. Some systems, such as Cone Trees or the Perspective Wall, allow the user to grab portions of the display and interactively drag the display to highlight a different segment of the display. Other systems, such as the Starfield displays, Bead, VIBE, InfoCrystal, LyberWorld, or VisDB allow the user to add or subtract search characteristics and to watch as the display changes accordingly. In most cases, the principles of direct manipulation are applied so that sliders or direct dragging is applied to modify the display. This allows the user to explore without having to know the exact values that are stored in the database.

All of the systems allow the user to highlight specific areas of the display. Many of the systems operate in the manner of fisheye views, in that they allow the user to focus on or emphasize a single point, and have the system change the display to emphasize that set of points. When this happens, the display is modified in some way to reflect the new importance placed on that particular point. In most cases, the display is not distorted, as is the case with classical fisheye views. The display can be rotated to bring the item into view, as with cone trees, or it can be skewed to emphasize a set of points, as with LyberWorld, or the whole visible area can move to a different area of the display space to bring forward different items, as with MUE.

Direct manipulation is a guiding principle for all of the techniques and systems described. In some cases, as with Starfield displays and the VisDB system, manipulation is accomplished via sliders that act on the visible data. In other cases, the display can be manipulated directly, as with cone trees, the Perspective Wall, or LyberWorld, where the display can be rotated or changed by dragging items of interest around in the display.

All systems perform dynamic queries so that any changes made to a control are reflected in the display immediately. In some cases, the display is rotated or distorted to emphasize certain items. In other cases, the number of items visible on the display is changed to highlight one subgroup over the full display. Unfortunately, the requirement to allow novice users to begin working with little training is violated by some of the systems. All the systems described in the section, “Graphical Views Into Large Databases,” require training. InfoCrystal, in particular, will likely require extensive training.

Visual information seeking, as narrowly defined by Shneiderman and others, specifically calls for the user of starfield displays. However, the characteristic of starfield displays is that they show all data points on the screen at one time. From that perspective, all the systems described here satisfy this requirement. In addition, all systems tie the manipulation of their controls to immediate action on the display.

It appears, therefore, that all the systems satisfy most of the criteria for good graphical information retrieval systems. How well do they overcome some of the limitations of traditional information retrieval systems as enumerated by Korfhage? (see Section 1 above)

Limitations of the Information Retrieval Systems

Many of these systems still present a limited set of documents. The user enters a list of search criteria, and the system returns a set of documents. The systems all display the documents in new and interesting ways, but the set of documents is still determined by the system. Even the Starfield systems, which display all the data points in the search space, are at present limited to a few thousand items.

There are a few systems that have the capability to allow the user to graphically explore beyond the narrow confines of the original search. Chief among these is the fisheye system described by Fowler, Fowler, & Wilson (1991), MUE, and LyberWorld. These systems appear to be designed to allow the user to go beyond the confines of the visible display and extend the search space throughout the whole database. GUIDO, Bead, and InfoCrystal also seem to have the capability to allow the user to graphically explore beyond the current display, but it is not clear that those capabilities are built into the current systems.

All the systems display the documents in a graphical display space. The positions of the documents in the display space indicate the systems evaluation of each document on some user-defined criteria.

All the systems display relationships between the documents returned, determined by the documents' positions in the display space.

The graphical displays attempt to give the user an indication of the position of a document in the corpus of documents searched. Documents closer to the center of focus are deemed most important, with documents farther from the focus being of lesser importance. However, unless the user is allowed to graphically extend the search beyond the set of returned documents, it is still difficult to tell exactly why one set of documents was selected and not another. The fisheye system of Fowler, Fowler, & Wilson (1991) does the best job of making the criteria for retrieval clear to the end user by clearly presenting the lines of relationship used to effect a search.

Whether the full document or only a bibliographic record is available is still system dependent. All the systems described here have the capability to return the full document if it is available.

It is still difficult for the end user to tell whether all relevant documents have been returned. The relationships between returned documents are clear, but not so clear is the relationship to documents not retrieved. Systems that allow the user to graphically explore further offer the most hope that the user will be able to find and retrieve other relevant documents that were not retrieved during the initial search probes. If setup correctly, Starfield displays and VisDB, with their reliance on sliders to set the parameters for retrieval, can make this process clearer. However, Starfield displays are limited to criteria set by the designers of the systems. VisDB relies on the user to set up the criteria for the sliders, so that unless the user knows exactly what is in the system, it is not possible to be certain that the correct search parameters are being used.

All the systems allow the user to customize the display by changing the focus of the system. The user can change the focus to any point on the display of interest and, in most systems, can hide or show more information depending on individual needs.

Because all the systems allow some graphical manipulation of the system, manipulating the system changes the weight of the query automatically. As mentioned earlier, however, only a few systems allow the user to graphically slide out towards and beyond the edge of the system and have the query change and extend the search space automatically as the focus moves. The user does not even need to know that a different query is being formulated, although a continuous display of search characteristics is displayed in a separate window so that the user can monitor the search.

In all systems, the user has a large part to play in exploring the system and tuning it to display the information in a way that best suits the individual user. Again, for most systems, direct manipulation is limited to the set of points returned by a traditional search query. However, the user can usually manipulate those points in a wide variety of ways.

Although no one system adequately addresses all concerns, the graphical approach appears to offer powerful advantages over traditional text-based retrieval lists. In addition to the information provided by text-based lists, graphical systems provide the capability to display more information and to manipulate the information in such a way that our visual processing system plays a part in the information filtering process.

Conclusion

None of the systems described here is the perfect information retrieval system. Even with the best of these systems, the searcher can still never be sure that all relevant documents have been retrieved. The graphical systems do, however, make it easier to filter the set of returned documents to determine which documents might be relevant to the searcher's needs by making the display of relationships clearer and by making it easier to explore those relationships. In addition, the systems with the most power to influence the relevance of returned documents also appear to be those that are hardest to use. Starfield displays come closest to combining power with ease of use, but it is not clear that they can be expanded to handle large databases.

Each of the systems described here has strengths and weaknesses when considered for a role as a graphical front-end to an information retrieval system. I have abstracted the following characteristics as being among those that should be incorporated into a graphical information retrieval system:

- The system should be visual and graphical.
- Documents in the system should be represented by icons that can be different sizes, shapes, colors, and/or have sounds associated with them.
- The system should be based on the fisheye principle. When the user focuses on a single point, other points in the system should dim in proportion to their relationship to the focus point.
- The display of information should potentially be three-dimensional. The third dimension gives the capability to display relationships that cannot be seen in a simple flat display. This recommendation is tempered with the realization that adding a third dimension adds to the complexity of the system.
- The system should allow the user to move and rotate the display on the screen, bringing different parts of the display to the front on command. It should be possible to highlight several different parts at once to explore the relationships between disparate parts of the display.
- The system should allow the user to explore beyond the confines of the original search, by bringing parts of the display at the periphery to the center and thereby extending the search into previously unknown regions as new pieces of the document space are brought into the display.
- The system should allow the user to view relationships between several different kinds of objects. At a minimum, the user should be able to change the display to display relationships between search terms or to display relationships between actual documents.
- The system should allow the user to select and retrieve any number of documents by simply

pointing to the document in the search space and issuing a simple command.

- The system should be composed of several parts, consisting of the relationship display space, a partial list of returned documents, the user's original query, a display showing the initial system derived search request, and a display showing the current system derived search request.
- The system should allow the user several different ways to look at the same data set.

Would an information retrieval system based on all these characteristics be the ideal system? Probably not, but the power that these systems give to the individual user to control the direction and extent of the search, probably means that they come closer than any non-graphical system in use today.

References

Ahlberg, C., & Shneiderman, B. (1994a). The Alphaslider: A compact and rapid selector. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 365-371). New York: ACM.

Ahlberg, C., & Shneiderman, B. (1994b). Visual information seeking: Tight coupling of dynamic query filters with Starfield displays. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 313-317). New York: ACM.

Ahlberg, C., Williamson, C, & Shneiderman, B. (1992). Dynamic queries for information exploration: An implementation and evaluation. In Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems, (pp. 619-626). New York: ACM.

Brown, M. H., Meehan, J. R., & Sarkar, M. (1993). Browsing graphs using a fisheye view. In Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems, (p. 516). New York: ACM.

Card, S. K., Pirolli, P., & Mackinlay, J. D. (1994). The cost-of-knowledge characteristic function: Display evaluation for direct-walk dynamic information visualizations. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 238-244). New York: ACM.

Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The Information Visualizer, an information workspace. In Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems, (pp. 181-188). New York: ACM.

Chalmers, M., & Chitson, P. (1992). Bead: Explorations in information visualization. In Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 330-337). New York: ACM.

Chimera, R. (1992). In Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems, (pp. 293-294). New York: ACM.

Fowler, R. H., Fowler, W. A. L., & Wilson, B. A. (1991). Integrating query, thesaurus, and documents through a common visual representation. In Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 142-151). New York: ACM.

Furnas, G. W. (1986). Generalized fisheye views. In Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems, (pp. 16-23). New York: ACM.

Hemmje, M., Kunkel, C., & Willett, A. (1994) LyberWorld - A visualization user interface supporting fulltext retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 249-259). London: Springer-Verlag.

Johnson, B. (1992). TreeViz: Treemap visualization of hierarchically structured information. In Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems, New York: ACM, pp. 369-370.

Johnson, B., & Shneiderman, B. (1991). Tree-Maps: A space-filling approach to visualization of hierarchical information structures. In Proceedings IEEE Visualization '91, (pp. 284-291). San Diego: IEEE.

Keim, D. A., Kriegel, H, & Seidl, T. (1993). Visual feedback in querying large databases. In Proceedings of Visualization ‘93, (pp. 158-165). Los Alamitos, CA: IEEE Computer Society Press.

Korfhage, R. R. (1991). To see, or not to see - Is that the query? In Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 173-179). New York: ACM.

Mackinlay, J. D., Robertson, G. G., & Card, S. K. (1991). The Perspective Wall: Detail and context smoothly integrated. In Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems, (pp. 173-179). New York: ACM.

Olsen, K. A., Korfhage, R. R., Sochats, K. M., Spring, M. B., & Williams, J. G. (1993). Visualization of a document collection: The VIBE system. Information Processing & Management, 29(1), 69-81.

Plaisant, C., & Jain, V. (1994). Dynamaps: Dynamic queries on a health statistics atlas. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 439-440). New York: ACM.

Rao, R., & Card, S. K. (1994). The Table Lens: Merging graphical and symbolic representations in and interactive focus+context visualization for tabular information. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 318-322). New York: ACM.

Rao, R., Card, S. K., Jellinek, H. D., Mackinlay, J. D., & Robertson, G. G. (1992). The Information Grid: A framework for information retrieval and retrieval-centered applications. In Proceedings of UIST’92 Conference of the ACM SIGGRAPH Symposium on User Interface Software and Technology, (pp. 23-32). New York: ACM.

Rao, R., Card, S. K., Johnson, W., Klotz, L., & Trigg, R. H. (1994). Protofoil: Storing and finding the information worker’s paper documents in an electronic file cabinet. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 180-185). New York: ACM.

Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). Cone Trees: Animated 3D visualizations of hierarchical information. In Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems, (pp. 189-194). New York: ACM.

Sarkar, M., & Brown, M. H. (1992). Graphical fisheye views of graphs. In Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems, (pp. 83-91). New York: ACM.

Spoerri, A. (1993). InfoCrystal: A visual tool for information retrieval. In Proceedings of Visualization ‘93, (pp. 150-157). Los Alamitos, CA: IEEE Computer Society Press.

Stone, M. C., Fishkin, K., & Bier, E. A. (1994). The movable filter as a user interface tool. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 306-312). New York: ACM.

Travers, M. (1989). A visual representation for knowledge structures In Hypertext ‘89 Proceedings, (pp. 147-158). New York: ACM.

Turo, D. (1994). Hierarchical visualization with Treemaps: Making sense of pro basketball data. In Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems, (pp. 441-442). New York: ACM.

Williamson, C., & Shneiderman, B. (1992). The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 249-259). London: Springer-Verlag.