Visualizing Collections of Information by People, Topic, and Time

Jeremy Goecks[1], Edward Cutrell[2], Susan Dumais[2], and George Robertson[2]

GVU Center[1]
College of Computing, Georgia Tech
Atlanta, GA 30332, USA
+1 404 385 1102
jeremy@cc.gatech.edu

Microsoft Research[2]
One Microsoft Way
Redmond, WA 98052, USA
+1 425 706 8259
{cutrell, sdumais, ggr}@microsoft.com

Abstract

Most information retrieval systems return a list of results in response to a user query. Many tasks, such as understanding which topics are active or who are the key people associated with topics, can better be addressed with a higher-level summary showing important trends and correlations in information creation and use. We describe the design and evaluation of GridViz, an information visualization that enables users to perform these tasks. GridViz uses a two-dimensional grid to visualize a large information collection by people, topics, and time. Cell entries encode information about the frequency and recency of document activity, including email, files, and shared information on collaborative web sites. In a user study we compared a traditional list view of results with the grid visualization for a variety of information access tasks. Users completed tasks faster with the grid visualization than a standard columnar list-view of the results, but there was no reliable preference for one view over the other.

Key words: Information visualization, information collection summary, grid visualization, metadata.

1 Introduction

One consequence of today's digital age is that more people are now being asked to make sense of large collections of information. Advances in technology and shifts in social behavior are increasingly placing office workers and home users in situations where they must work with large information collections.

Email is a dominant form of communication today. Not only do people receive email from friends, family, and colleagues, but they also receive email via mailing lists. People often feel overwhelmed by the amount of email that they receive, and there are no signs that this problem will subside [17]. Internet search engines often return thousands of documents in response to a user's query, and users have difficulty finding documents that interest them in these long lists of returned items [2]. Collaborative websites [3] are the newest space in which information collections are growing to an unwieldy size. These websites enable a group of people to build and maintain a collection of documents very easily by allowing anyone in the group to add, remove, or modify documents on the website.

Large information collections are difficult to interact with because most interfaces to these collections limit the ways users can view the collection. Browsing interfaces allow drill-down navigation along one or more pre-defined organizations but seldom provide useful summary information. Search interfaces are typically long lists of documents (as with web search results), or more detailed list views showing additional attributes in columns (as in mail readers).

List views make it difficult to obtain summary information about a collection and identify important trends in the collection. For instance, a worker cannot easily identify which of his colleagues has been sending him information about a particular topic, or determine which mailing lists have been the most active recently and thus bear watching. In a typical list view of web search results, users cannot easily determine whether the results represent a small number of principal topics or are spread throughout many diverse topics. In addition, identifying trends in a collaborative website becomes very difficult when a group is particularly active or large because it is difficult to determine what changes have been made to the website since it was last viewed.

The tasks just described require that users be able to summarize an information collection and operate on the summary. Users must identify trends, distributions of document attributes, and correlations among

attributes in the collection. People, topics, and time are particularly important attributes in an information collection, and many summary tasks concern relationships between these attributes.

We have developed GridViz, an information visualization that enables users to view summaries of an information collection. These summaries are based on the people, topics, and temporal attributes of the collection. GridViz displays the query results from an information retrieval system that indexes documents that a user is interested in. This includes the user's email, the documents he has worked on, and the web pages and collaborative websites that he has visited. Visualizing such document collections is an especially challenging problem because the collection can be very large. For instance, the query 'chi' on one author's laptop machine returned 1400 documents.

GridViz uses a two-dimensional grid to visualize a large information collection by people, topics, and time. Cells encode information about frequency and recency of document activity, including email, files, and shared information on collaborative web sites.

In the next section we discuss work related to GridViz. We then describe GridViz in detail, and present a user study that compares the visualization to a list view presentation. We follow with directions for future investigation and a summary of the contributions of this work.

2 Related Work

Our work draws inspiration from previous work in three areas: (1) visualizations for information collections; (2) uses of metadata in the user interface; and (3) visualizations that employ a grid motif. We present related work in each area and draw distinctions between this work and GridViz.

2.1 Visualizing Information Collections

Much research has been devoted to visualizing information collections. In Visual Information Seeking [1], users employ dynamic query filters and a Starfield display to search an information collection. The Starfield display is a spatial display, and the collection's elements are plotted on the display. Users query the collection via the filters, and elements that meet the query are visualized on the display in real time. For the FilmFinder application described, the x-axis displayed the film's production year and the y-axis displayed the film's popularity. For this application, the axes were fixed.

The Envision system [13] visualizes query results from a digital library of computer science literature. Envision uses a two-dimensional scatter plot to visualize returned documents; users can choose which document attributes to place on each axis. Possible attributes are document relevance, author names, index terms, document type, and publication year. An icon represents each document on the plot, and users can configure the icon's color, shape, or size to encode the document's relevance.

ThemeRiver [5] uses a river metaphor to visualize the thematic composition of a large document collection over time. ThemeRiver uses a river metaphor and encodes time, collection themes, and theme strengths using various attributes of a river. The river's flow enables users to view the overall thematic composition of the collection and follow a particular theme in the collection through time.

These applications illustrate the space explored by visualizations. We have integrated ideas from each of these applications into GridViz. GridViz uses a two-dimensional grid that is more sophisticated yet in the same spirit as a scatter plot. Our visualization also summarizes a document collection in order to show trends in the collection. In contrast to ThemeRiver, though, it summarizes multiple aspects of the collection so as to enable users to identify trends and correlations beyond just themes.

2.2 Metadata in the Interface

Previous work has used metadata in the user interface for various purposes. Hearst et al. [5] developed a novel user interface for web searching that utilizes metadata to help users navigate large information spaces and more easily find information that they are looking for. When the user executes a web search, metadata categories that match the search query are returned in addition to documents that match the query. Metadata categories contain documents whose metadata corresponds to the category and additional categories that are related to category.

FotoFile [11] is a multimedia organization, storage and retrieval system that uses metadata to help users organize their content. FotoFile uses a hierarchical folder structure that is determined by the metadata of content stored in the structure. For instance, there might be a "places" folder in the hierarchy, and a sub-folder called "Fort Lauderdale." Content is stored in the folders based on the content's metadata. FotoFile also uses a hyperbolic tree to enable users to browse stored content via the metadata networks.

Shniederman's hieraxes system uses a two-dimensional display that shows categorical and hierarchical attributes on the axes with individual search results shown in the cells [16]. Users can refine

their searches by selecting folder icons to reveal sub-topics and can filter the results by category.

These systems all employ metadata to enable the user to better understand a collection of information. However, the goals of each application are very different.

2.3 The Grid Motif in Visualizations

As discussed above, Envision uses a grid motif to display query results. Two other prototypes have used grids to represent email patterns. NEC's VisualMail [12] shows each message on a two-dimensional grid whose axes are time and content. VisualMail displays the whole message on the grid. TimeStore [9] creates a grid by placing people on the X axis and time on the Y axis. Users can view 3 months, a month, a week, or a day at a time. A circle in a cell indicates that the person has sent email to the user during that time period; the circle's size indicates the amount of email sent.

InfoGrid [14] uses a grid to display thumbnail images of documents retrieved from a query; the query results are displayed in relevance order on the grid. Hence, InfoGrid encodes information in only one dimension of the grid. In contrast, GridViz encodes information in two dimensions directly using the grid's axes and in two other dimensions indirectly.

GridViz provides richer and more flexible summary options than the above visualizations. GridViz visualizes a document collection using people, topics (content) and time. People are further subdivided into authors (or "from: people"), "to: people," and "cc: people" to support more advanced visualizations. GridViz enables users to choose which attribute to assign to the grid's axes, and allows users to sort the grid in multiple ways.

Parallel Coordinates [6] uses a grid to show multivariate/multidimensional relationships. The X axis is used for variables, while the Y axis shows their values. The user sees patterns where variables are correlated or uncorrelated. GridViz also allows the user to see correlations, but in more detail for two selected dimensions (people, topics, or time).

3 GridViz

Our goal in designing GridViz was to provide users with an interface that enabled them to explore trends, correlations, and other relationship in a large information set. Such information sets could arise as the result of a search, or as a standing profile monitoring sites and sources of interest.

The following examples more concretely illustrate the kinds of tasks we wish to better support. One of our colleagues recently needed to send email to everyone involved with a newly formed research initiative. Because the initiative was new, there was no mailing list for the initiative. A visualization that showed him which people, distribution lists, and collaborative websites are associated with the topic of interest would have greatly aided him in finding the right people. Consider another example. A group manager wants to touch base with everyone in her group on a regular basis, and thus she would benefit from a visualization that showed her when she last sent email to each person in her group.

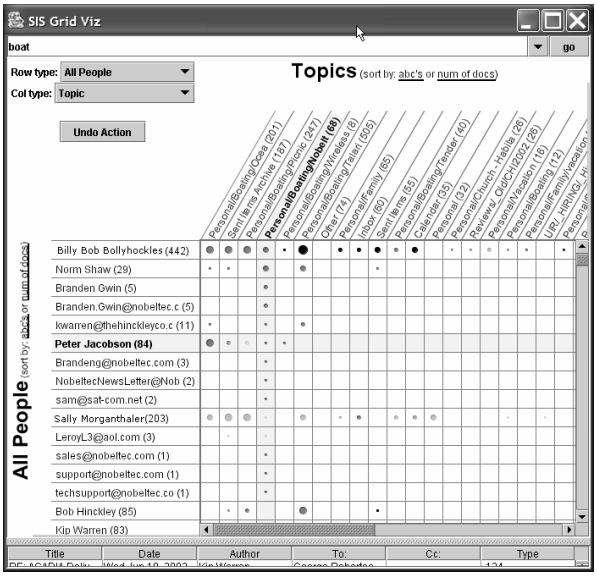

Figure 1*. GridViz visualization (anonymized).*

These two examples also illustrate two critical design requirements of GridViz:

1. *Use people, topics and time to visualize the collection.*
2. *Make the visualization highly flexible.*

3.1 Visualization Overview

GridViz uses a grid motif to show relationships between people, topics, and time (Figure 1). Users assign one of these attributes to the X axis and another attribute to the Y axis. Topics are assigned to the Y axis and people are assigned to the X axis by default. By virtue of the attributes assigned to the grid's axes, each cell in the grid represents two attribute values. Because email is such an important document type, the visualization also enables users to place people in the "to:" (or cc: or from:) line of emails on the grid axis as well. Using these subgroups enables users to begin to investigate social patterns present in the collection.

Recall that GridViz visualizes the document collection returned when the user queries an information retrieval system that has indexed documents of interest to the user. When the user performs a query, GridViz obtains the results from the query and processes the documents in order to obtain the necessary information to populate the grid. GridViz creates a list of all the people, topics, and time periods that appear in the collection, determines the number of documents associated with each attribute value, and populates the grid.

On each axis, GridViz lists all attribute values that were found in the document set for the attribute assigned to the axis. (E.g. in Figure 1, people are assigned to the Y axis; hence, all the people in the document set are listed on the Y axis.) In parentheses beside each value is the total number of documents that share that value.

Initially, attribute values are ordered on an axis based on the number of documents that share the value; values that have the most associated documents are listed first. In Figure 1, there are 85 documents associated with the person Bob Hinckley and there are 505 documents associated with the topic 'Personal/Boating/Talari.'

Documents returned from the query are placed in grid cells based on their attribute values. Documents are often placed in more than one grid cell; for instance, an email message often has multiple people associated with it. A circle in a grid cell indicates that there are documents associated with that cell. The size of the circle corresponds to the number of documents in the cell. The circle's saturation is based on the timestamp of the most recent document in the cells. Hence, faded circles indicate that the cell does not contain any recent documents, and bright red circles denote the presence of recent documents.

The circles in the cells are interactive. When the user mouses over a circle, flyover text appears which

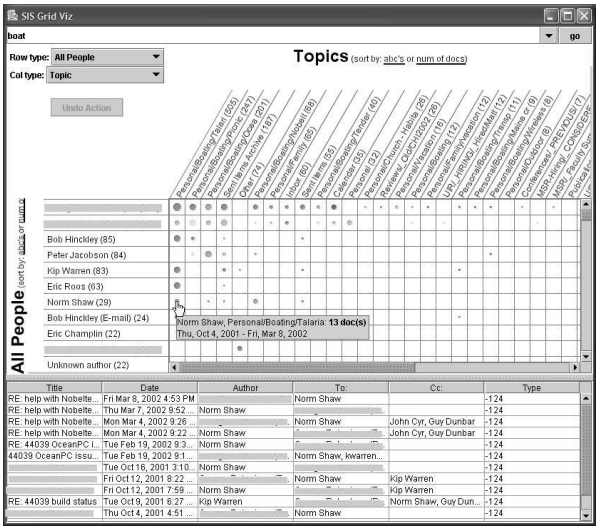

***Figure 2**. Mouse-over text for a circle; when user clicks on circle, documents are displayed below.*

indicates the attribute values associated with the cell, the number of documents in the cell, and the earliest and latest timestamp of the documents. If the user clicks on a circle, the cell's documents are displayed in the list view below the visualization (Figure 2).

Users can sort the grid in several different ways to better view particular relationships in the collection. Hyperlinks beside the axis labels show the sort alternatives. Frequency is the default. For people and topics, another sorting is alphabetical; for time, another sorting is by recency. Users can also reorder the columns relative to a particular row. Clicking a row sorts the columns based on the number of documents in the row's cells (Figure 3). For example, if people are assigned to the X axis and topics are assigned to the Y axis, then the user can click on a person (a row) to sort the topics (the columns) based on the number of documents in each topic associated with the person; this sort shows which topics a person is most closely

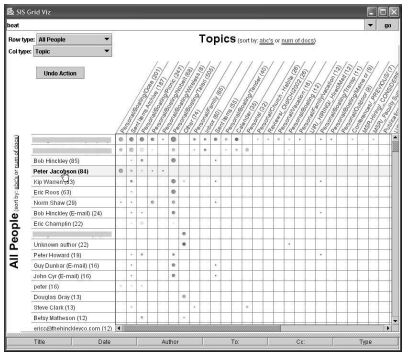

***Figure 3**. Columns sorted by row.*

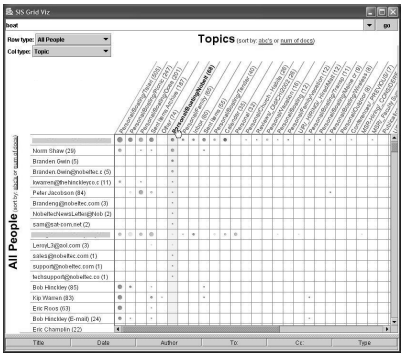

***Figure 4**. Rows sorted by column.*

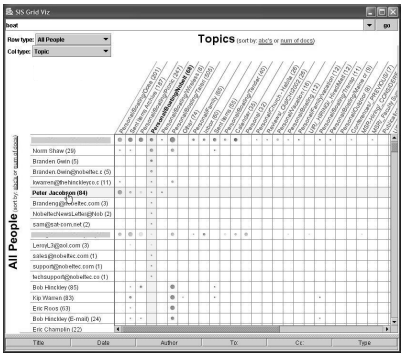

***Figure 5**. Columns sorted by row; rows sorted by column.*

associated with. It is also possible to sort the grid's rows by clicking on a column (Figure 4). The grid can also be sorted both by a row and a column (Figure 5).

By changing the attributes associated with the grid axes, users can view different summaries of the document collection and explore different relationships among attributes. In sum, the user can view twenty different summaries of the collection. The ability to view so many different summaries, coupled with the sorting features discussed previously, makes GridViz a very powerful and flexible tool for exploring higher level qualities of a document collection.

3.2 Implementation

We used Java Swing [8] and Java 2D [6] to implement the GridViz. We used Java 2D to draw the grid's labels, cells, and circles; we used Java Swing to implement the other components of the interface and manage the application's layout.

The GridViz visualization gets its input from an information retrieval system. The system needs to provide search results for a query, along with metadata about each. For the experiments described below, we used metadata representing time, person (to:, from: and cc:), and topic.

Creating a data structure that enabled GridViz to respond to user's actions in real time was the most challenging implementation issue. The key issue for the data structure concerned what information to store in memory and what information to create on the fly.

Sorting can be done quickly once the grid has been created, but creating the grid is the slowest operation in our interface. Storing all possible (or even common) grids in memory wasn't feasible because the grids would require too much memory. Hence, we optimized the data structure so that it would be able to create a grid quickly. When the user performed a query, we performed a significant amount of preprocessing on the returned documents; during preprocessing we found and stored all the people, topics, and time periods for a document set. Once we had this data, we could create the grid quickly. Creating a new grid is fast; even for document sets that have 5,000-10,000 documents, GridViz can create and display the grid in approximately 1-2 seconds.

GridViz did not perform text analysis to obtain the topics in the returned document set. Instead, we used mail folders as a simple approximation for topics; each folder and each subfolder in a user's mailbox represented a different topic in the visualization. In our workplace, the great majority of people do use folders for this purpose, and hence approximating topics this way is acceptable in our environment. In the future we would like to implement a topic identification component for GridViz; and there are known techniques to do this [10][15].

4 USER STUDY

To aid in the evaluation of the interface, we performed a user study in which participants had to answer a number of questions using both the GridViz interface and a standard list-view interface. After finishing the experimental tasks, participants answered a questionnaire about their experience.

4.1 Methods

Participants

Nine adult coworkers (3 female and 6 male software professionals) between the ages of 25 and 44 were recruited for this study. No participants had any experience with the GridViz interface. All participants used list view interfaces such as those in Outlook on a daily basis, and five had some experience using the particular list-view interface tested in the experiment.

Procedure

The experiment was divided into two halves. Participants used one interface in the first half and another interface in the second. The order of presentation was counterbalanced across subjects. Before each half, participants were given a short tutorial on the use of the interface and were encouraged to interact with it before executing two practice tasks. Users performed 20 tasks in each half, for a total of 40 tasks. At the end of the experiment, participants completed an online questionnaire. The

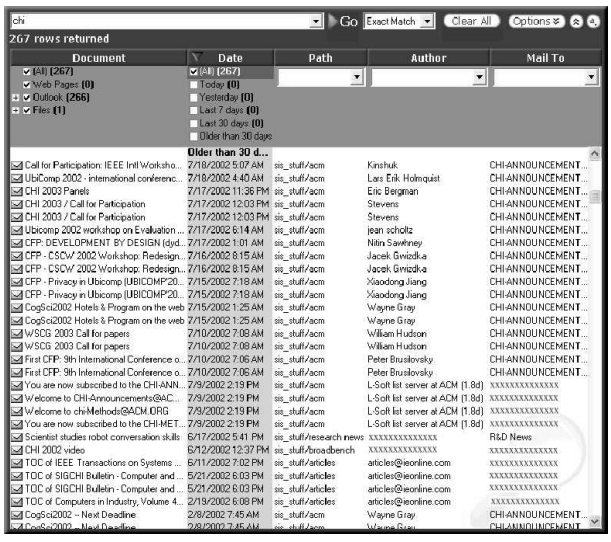

Figure 6*. List-view interface used in study.*

total time for the experiment was about 90 minutes.

During the experiment, participants referred to a spreadsheet that was divided into two sections (for each half of the experiment). The spreadsheet comprised 3 columns: The first contained the query word that participants were to type into the interface to initiate a search (e.g., *ACM*). The second column contained the question to be answered (e.g., *who sends the most mail about ACM?*). The third was the answer column into which participants typed their answers. Participants were encouraged to copy and paste the queries from the spreadsheet to prevent typing errors, but this was not rigorously enforced.

For both interfaces, participants began a trial by entering the query into the interface and executing a search. Because different queries took different times to execute and render, the beginning time for each trial was set to when the result set was fully rendered on the screen. When participants found an answer they spoke it aloud and typed it into the answer column of the spreadsheet. They were given no feedback as to whether their answer was correct. The time the answer was spoken was taken as the ending time for that trial.

List-view Interface

The comparison interface used for this study was a fairly traditional list view interface (Figure 6) in which each matching document is listed on a line and the properties of that document are displayed in columns (e.g., name, date, author, etc.). By default, results were sorted by time, but clicking on any column header would sort by that column. In addition, it was possible to filter based on specific values in any given column.

Tasks

A database of over 6000 emails to distribution lists, labeled with the list category, was used for all queries in this experiment.

We developed forty search-based tasks requiring some sense of overview information such as time, frequency or a combination of the two. Example questions included: *Who are the 4 most active contributors to the Neuroscience topic? What topic has Krishna Jones sent mail about most recently?* Each question could be answered from the information returned by the client, although some were more difficult and required more interaction with the visualization than others. To ensure that results from different participants were comparable, we fixed the keywords for each query, and all queries were performed against the same database of email. This insured that each participant received the same results for the same query. The number of items returned by each query varied between 20 and 2209, with an average of 360 documents returned.

All participants performed the same 40 search tasks. They used one interface for the first 20 tasks and another for the remaining 20. The order in which participants saw the interfaces was counterbalanced across participants. Queries were also counterbalanced. Because of an equipment failure and scheduling problem, one participant was able to complete only 12 of the 20 tasks in one condition. Analyses were performed both with and without this participant. Since there was very little difference in the analysis, and the timing data were very similar to all other participants, we elected to include it in the analyses we report here.

4.2 Results

The main independent variable in the experiment was the interface used. All interface comparisons were made between subjects because the main source of timing variance was the task. We analyzed both the time it took participants to finish each task and a variety of subjective questionnaire measures.

Accuracy

Accuracy was just over 80% for both interfaces. Participants were allowed to give up at any time during a trial if they could not find an answer. After 3 minutes had elapsed for a task, participants were encouraged to move to the next task. Some participants continued searching, but most gave up at this time. Participants gave up on a total of 6 trials, and all of these were in the list-view condition. Half were due to one particularly difficult query.

Task Completion Time

Mean log completion times were used in the statistical analyses to normalize the common skewing and variability associated with response time data. However, because log times are hard to understand, we show raw completion times in the figures reported below to give the reader a sense for the magnitude of the effects. Figure 7 shows the mean completion time associated with each condition.

We performed a 2 (GridViz vs. List-view) x 40 (Task) Analysis of Variance ANOVA. The GridViz interface was significantly faster than the List-view interface $F(1,195)=7.87$, $p<0.01$ (see Figure 7). In spite of the fact that participants had a great deal of experience with list views in general and several had used the particular List-view interface tested in the experiment, GridViz was reliably faster. There was also a significant effect for Task, $F(39,195)=4.13$,

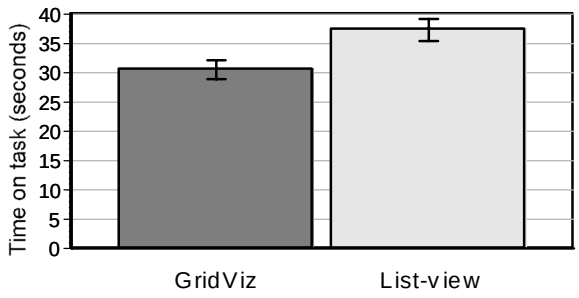

Figure 7. *Mean task completion time.*

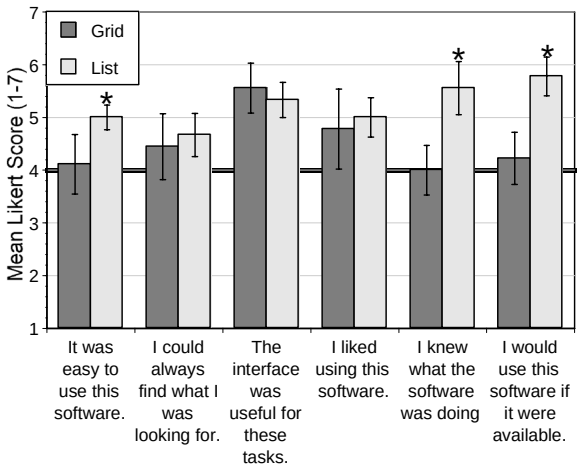

Figure 8. *Questionnaire scores for overall quality and preference (* are significant).*

$p<0.01$, which is not surprising. Some tasks were more complex than others and required additional time. In addition, there was a significant interaction between Task and Interface, $F(38,195)=2.43$, $p<0.01$. That is, performance for the interfaces depended on the task.

This analysis revealed two main points: First, the GridViz interface was overall faster than the traditional List-view interface; and second, this advantage was not uniform across all tasks. All the tasks in which the List-view was faster than the GridViz asked questions about the time of items, e.g., "What was the first message sent to the Ipaq discussion group?" Such questions can be answered quickly in the List-view by simply sorting by time which was the default sort. For GridViz, the default presentation was People by Topic, so additional actions were required to sort by time. Inspection of the hue provides some cues about recency, but they are not sufficient to identify the most recent item.

Subjective questionnaire measures

After the experiment, participants completed a brief online questionnaire. The questionnaire collected information regarding common email usage, ratings of the two interfaces (on a 7-point scale), and open-ended questions about the best and worst aspects of each interface. Although participants performed the tasks faster with the GridViz interface, they actually seemed to slightly prefer using the List-view. On 3 of the 6 questions about the overall quality of the interface, scores were significantly better for the List-view than the GridViz interface and these are marked with an asterisk in Figure 8. For the remaining 3 questions, there was no reliable difference in preference between the two interfaces.

5 Future Work

We would like to study how people use the grid visualization during their daily tasks. Although the study that we performed showed that users can perform some tasks better with the grid visualization, the study does not tell us how, when, or in what context people would use the visualization on a day-to-day basis. In addition, the results from the subjective questionnaire suggest that people were somewhat uncomfortable with GridViz's presentation of an information collection. We hypothesize that users would adjust to GridViz's presentation if they used it on a regular basis and became familiar with it. We plan to deploy our visualization to a sizable user population and study these questions.

We would like to add the ability for users to return to previous states and use their interaction history to inform future actions. We would also like to explore other methods to display the documents in a grid cell. Currently, GridViz only encodes frequency and recency information, but we could encode additional attributes such as document type.

6 Contributions

GridViz provides a simple, intuitive interface for users. Users are very familiar with grids and plotting elements on a grid; thus they can easily understand GridViz's conceptual model. The grid layout, coupled with the colored circles in the cells, provides a perceptual summary of the information collection. This summary enables users to quickly identify correlations and trends between the attributes assigned to the grid's axes. It is easy for users to find the cells that have large densities of documents, and therefore which attribute values are strongly correlated in the collection. Sorting the grid's rows/columns or sorting the columns by row (or vice versa) is simple to do and makes other relationships among attribute values visible. The user can readily assign different attributes to the grid axes and summarize the collection using these attributes.

Despite the simplicity of GridViz's interface, it is a very powerful and flexible tool for exploring large information collections. GridViz visualizes a document collection using the most salient document attributes – people, topic, and time; in addition, the visualization further breaks down people into authors (or "from: people"), "to: people", and "cc: people" in order to better visualize email. With GridViz, users can view the overall relationship between two attributes in a collection, or they can explore the ordered relationship between any single attribute value (e.g. a person, a topic) and all the values of another attribute by sorting the grid using that value.

An experiment showed that users completed trend and correlation-based tasks more quickly with GridViz than with a standard list-view interface. This was true in spite of the fact that participants are all familiar with and use list-view interfaces on a daily basis. Preferences were mixed with participants feeling more confident using the list-view.

In sum, GridViz is a simple yet powerful tool that enables users to explore and understand information collections in a qualitatively different way than they can with current interfaces.

References

[1] Alberg, C. and Shneiderman, B. (1994). Visual information seeking: tight coupling of dynamic query filters with starfield displays. *Proceedings of CHI '94*, Boston, MA, p. 313-317.

[2] Dumais, S. T., Cutrell E. and Chen, H. (2001). Bringing order to the web: Optimizing search by showing results in context. *Proceedings of CHI '01*, Seattle, WA, p. 277-283.

[3] Guzdial, M., Rick, J., Kerimbaev, B. (2000). Recognizing and supporting roles in CSCW. *Proceedings of CSCW 2000*, Philadelphia, PA, p. 261-268.

[4] Havre, S., Hetzler, E., Whitney, P., Nowell, L. (2002). ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 8, No. 1, January-March 2002, p. 9-20.

[5] Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K. (2002). Finding the flow in Web site search. *Communications of the ACM*, 45 (9), Sept. 2002, p. 42-49.

[6] Inselberg, A., and Dimsdale, B. Parallel Coordinates (1990): A Tool for Visualizing Multidimensional Geometry. *Proc. Of IEE Conf. on Vis. '90*, Los Alamitos, CA, 361-378.

[7] Java 2D. http://java.sun.com/products/java-media/2D/

[8] Java Swing. http://java.sun.com/products/jfc/

[9] Jovicic, S., and Baecker, R. (1999). Time-based archiving and retrieval of email. *Workshop on History-Keeping in Computer Applications*, Human-Computer Interaction Laboratory, University of Maryland. December, 1999.

[10] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 (62), 1982, p. 59-69.

[11] Kuchinsky, A., Pering, C., Creech, M., Freeze, D., Serra, B., and Gwizdka, J (1999). FotoFile: A consumer multimedia organization and retrieval system, *Proceeding of CHI 99*, Pittsburgh, Pennsylvania, p. 496-503.

[12] Kudo, M.,Tanaka, M., and Koseki, Y. (1997). Information Visualization for electronic mail management. *Proceeedings of Visual 97*, San Diego, CA.

[13] Nowell, L., France, R., Hix, D., Heath, L., Fox, E. (1996). Visualizing search results: Some alternatives to query-document similarity. *Proceeding of SIGIR 96*, Zurich, Switzerland, p. 67-75.

[14] Rao, R., Card, S., Jellinek, H., Mackinlay, J., and Robertson, G. (1992). The Information Grid: A framework for information retrieval and retrieval-centered applications. *Proceedings of UIST 92*, Monterey, CA, p. 23-32.

[15] Salton, G. (1991). Developments in automatic text retrieval. *Science*, Vol. 253, p. 974-980.

[16] Shneiderman, B., Feldman, D., Rose A., and Ferre Grau, X. (2000). Visualizing digital library search results with categorical and hierarchical axes. Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio TX, p. 57-66.

[17] Whittaker, S. and Sidner, C. (1996). Email overload: Exploring personal information management of email. *Proceedings of CHI 96*, Vancouver, BC, p. 276-283.