

Visualizing Evolving Searches with EvoBerry

Edward Suvanaphen and Jonathan C. Roberts
Computing Laboratory, University of Kent, UK
es45@kent.ac.uk, j.c.roberts@kent.ac.uk

Abstract

Studies show that roughly one-third of searches that are performed on the web require the user to initiate subsequent searches. Bates [1] theorized that with every search the user will encounter new information, which in turn leads to new ideas and directions. This process causes a change, not simply in the query terms being used but also to the nature of the information retrieval task itself; Bates called this the Evolving Search. She also noted that Evolving Searches utilize many different information sources, generate substantial quantities of data and require easy methods to save and recall data. Although current search tools are exceptionally efficient at locating highly ranked pages, the tools do not encourage or support the user in an evolving search. In this paper we present techniques that aid users to find, view and manage data produced from their evolving searches. In particular, we introduce the EvoBerry environment, which we have developed for use with evolving searches. EvoBerry includes methods to visualize additional search result information (such as length of page or file type), manage the user's session and browsing history, compare result sets, and store and bookmark items for future reference.

Keywords—Evolving searches, Search result visualization, Information Visualization, Berry-picking model.

1 Introduction

Everyday, millions of users log onto the Internet and search for information. In 2003 Google reported a staggering average of 250 million searches per day through their search engine. Search engines are extremely popular and are often the first destination for a web user. This view is supported by analysis from Alexa.com, where the top five most popular websites are all search engines. This phenomenon has generally encouraged developers to focus on the efficiency and accuracy of search algorithms. Developers aim to present the most relevant result as the first element in a rank ordered list, or at least on the first page. However, recent studies demonstrate that users are not always able to find information they require with a single search. A study by Jansen et al. [8] of search engine logs

showed that roughly one third of all searches performed were immediately followed by a second, third and fourth search.

Clearly, users change their keyword terms to create subsequent searches after viewing the results from the current terms. Various researchers have proposed models to explain this reformulation. The general theory is explained in the information access process model [6]. Users first have an 'information need' that gets expressed as a 'query'. The query is sent to the system, the results received and evaluated, finally the user reformulates the query terms on successive searches to achieve their goal. Marchionini [11] extends this basic model by proposing an iterative model of information seeking functions, where users can return to the problem definition at any step. Bates [1] suggested that by continually browsing returned information the user develops new ideas and directions for their search, and as a result *evolves* his/her information need and forages for information [13]. Bates [1] theorized that with each different conception of the query, the user would identify useful information and references. She named this *Evolving Searches*.

One interesting side-effect of the evolving search, was that the user's 'information need' would not be satisfied by a single retrieved set, but instead through a series of searches that would necessitate looking at different sources at each stage of the ever-modifying search. Bates named this the Berry-picking model; analogous to the picking of berries in the forest, where berries are scattered between bushes, forcing the picker to wander among the bushes picking one set of berries at a time. This analysis implies that the user's final answer is often a conjugation of several pieces of information obtained throughout the search, and as a result the information (past and present) acquired during the search is important to the user.

This paper presents the EvoBerry environment, which aids the user through evolving searches. EvoBerry has been designed to enable users to perform evolving searches based on the berry-picking model. The program was written in Java using the Swing Toolkit, the JDIC web-browser was used to render webpages and the Yahoo web-search API was used to gather search result data.

The environment contains three main window types: (1) the browser, (2) the results window and (3) the toolbox, (see Figure 1). These windows are all managed within a workspace. Users create searches by inputting their search terms into the search box at the top of the workspace; previous searches can be retrieved through a drop-down list. The search is then submitted to the Yahoo search engine through the Yahoo api, and a set of results are returned and displayed in the results window. Subsequently, these results are then processed to obtain webpage addresses, which are then visited and parsed to gain additional meta-data.

2 Towards Evolving Searches

Currently, various web search-result visualizations have been developed, from text displays, simple 2D plots ([10] [15]) to complex 3D worlds. There are a number of tools that visualize search results using graphical representations. SQWID [12] used a 2D plot where search results were located according to their relevance to various search terms. Cugini et al. [2] implemented a number of 3D tools for displaying search result data, from a simple 3D scatter plot, to a complex 3D globe. VR-VIBE [20] created a 3D space in which documents and keywords were located based on their relative relevance.

Interaction and manipulation is an important aspect of any visualization. With search result visualization there is a challenge to display a lot of information in a small space. Distortion techniques have been utilized that distort some detail of the display while showing the context in more detail. One technique associated with 2D and 3D visualization alike is the Fish-eye view technique. Originally implemented by Furnas [3], the fish-eye view could be used to display results in detail, while at the same time giving a broad overview of the data (albeit in less detail). Various researchers have used this technique in textual visualizations [9, 17, 14] and more recently in web search result visualizations [16].

Although, it is possible to perform evolving searches using existing tools, they do not effectively support this process. In a recent NSF workshop on ‘Creativity Support Tools’, Hewett et al. [7] explained that traditional developers have focused on fact-based and known-item searches, and that techniques should be developed to aid the user in more creative searching. Thus there is a need to develop systems that better manage and recall data, supporting the user in their exploratory search.

There are many real life examples where people use exploratory searches, and look through multiple databases and different sources, to achieve their goal. For example, for a recent conference we wished to look for a London hotel that was both cheaper than and within walk-

ing distance of the conference hotel. The following processes were used. First the conference website was visited, in order to gather keywords for the conference including: zip code, conference hotel name and location information. Then Google was used to search for ‘London hotels’, and some hotel pages were viewed. Expedia.co.uk was loaded from a bookmark, which had been saved from a previous session, and was used to find pricing and other possible hotels. Multimaps.co.uk was consulted to find location and distance information. This map gave extra keywords, and we realized that we may be able to get a cheaper hotel with a short trip on the underground train. Then the original conference website was re-consulted to find the closest underground station and the original search reformulated to change the geographic area of the required hotel. Finally, the rail route was then calculated to make sure the chosen hotel was accessible and Expedia was used to book the hotel.

This example demonstrates that the user’s search query changes, because their information need changed. They may need to refer to multiple sources and a user may need to refer back to previously viewed information (such as webpages, bookmarks, or previous searches). From this example, we notice that evolving searches have some important requirements:

1. Information Seeking. As with any search result visualization, the system needs to allow the user to choose which database or search engine they are to use, and enter and reformulate queries. The user needs to view the results, look at different information sources, and select important results.

2. Effective Visualization. The information needs to be presented to the user such that they can seek the information they require. Such visualization may be hard, as it is far too easy for the user to become overloaded with too much information and pertinent information becomes hidden. Hence it is important that information is presented to the user in an effective manner, where the user can discover interesting results quickly and effectively.

3. Information recall. Because the user’s information needs change they must be able to keep careful track of their data. Information that they may have gathered and discarded at the beginning of their search may later become important.

4. Information management. New ideas are likely to be generated during the evolution of the search, from different sources and different databases. They should be able to start on a trail, temporarily follow a new trail and then return to the original. There is a need to group-together concepts that are all from the same trail or session, to enable the user – as they change their overall information need – to keep track and return to previous sessions.

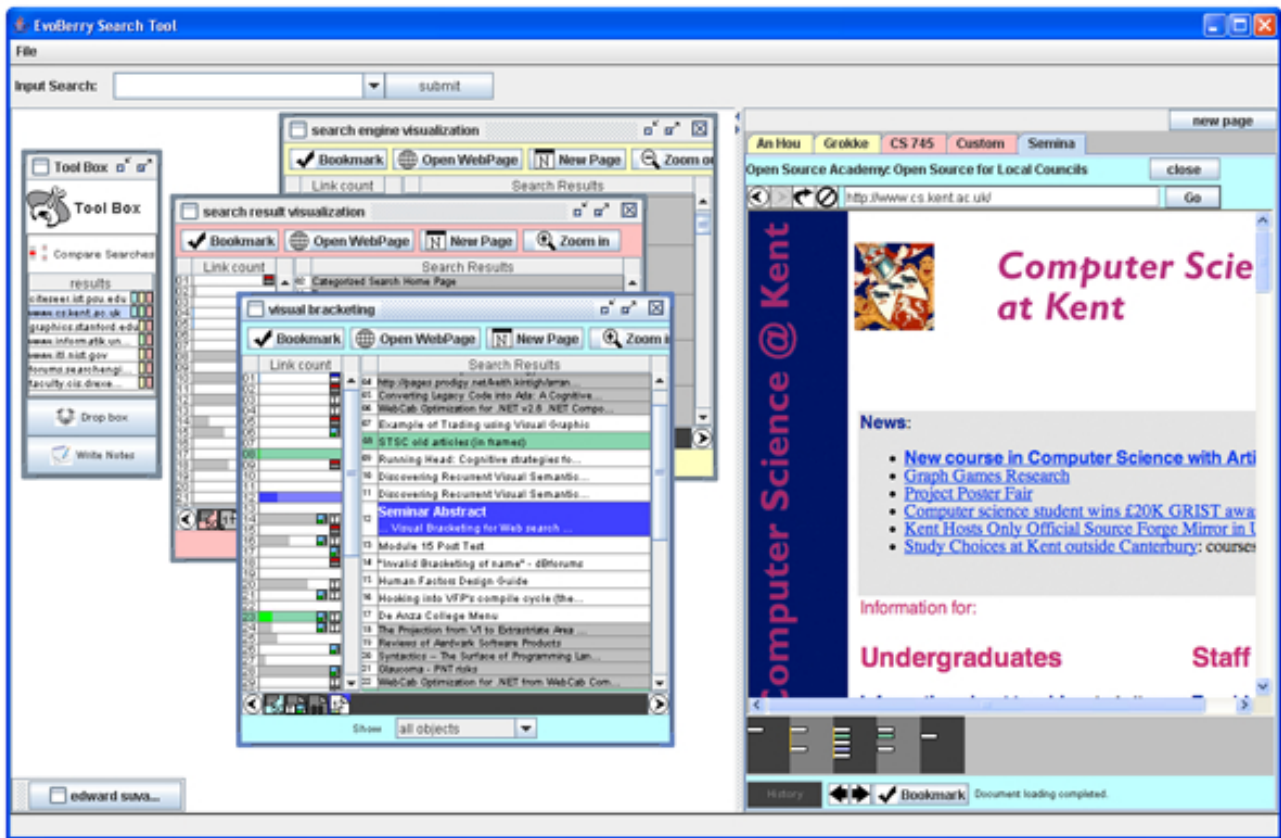


Figure 1: **The EvoBerry Environment:** On the far left is the *ToolBox* with the *Comparison view* open, center are three *Results windows*, and far right is the *Browser window* with five tabbed web pages (one in view). The comparison view shows all the web pages that exist in more than one of the searches. The browser window has its visual history open (bottom right) displaying the users page visits.

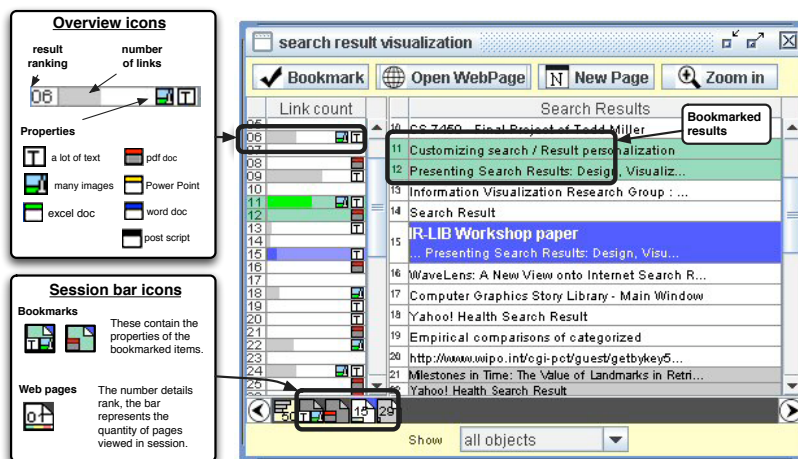


Figure 2: **Results window details:** this contains three main parts, the *overview* (left), the *main view* (center/right) and the *session bar* (bottom). There are two bookmarked results in the main view (numbered 11 and 12 and highlighted in green) and four session icons in the session bar (two bookmark icons and two web page icons).

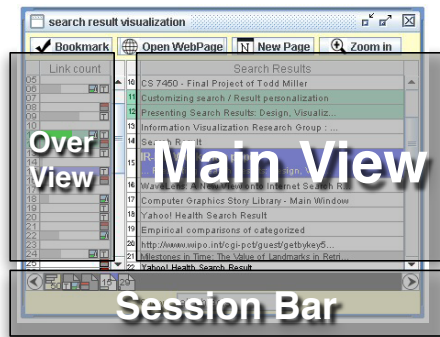


Figure 3: **Results window parts:** this contains three main parts, the *overview* (left), the *main view* (center/right) and the *session bar* (bottom).

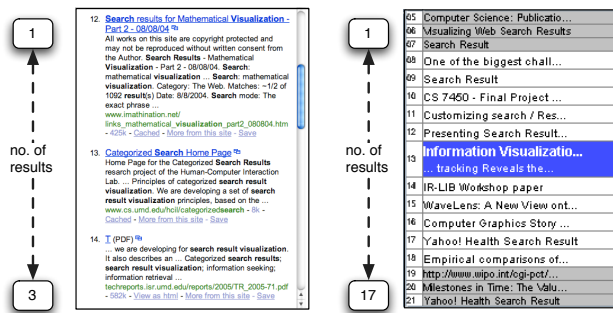


Figure 4: **The Visual bracketing technique.** On the left, the standard web browser view of a list of search results, on the right, a list of search results viewed using Visual bracketing.

3 EvoBerry: visualizing evolving-searches

We first describe how EvoBerry follows the four requirements, and then include some details of our evaluation.

3.1 Information seeking

The workspace begins with the ToolBox window already on the workspace. Whenever a search is performed a results window is added to the workspace. One strategy for effective information seeking is to present information that a user would have had to achieve cognitively; this could be to present summary information, comparison information or other meta-information. EvoBerry utilizes several summary mechanisms to allow the user to quickly find relevant information.

EvoBerry visualizes meta-data to help the user focus on relevant document types with various attributes. Various attributes can be calculated by parsing the webpages or accessed through a search engine API. For instance, information about the number of links, amount of images, word

frequency, where the key terms appear can all be calculated. In EvoBerry we include this information as icons.

The results window contains two linked panes, on the left is the overview, and the right is the main view, see Figure 3. The overview describes a rank ordered list of results in a summary form. This summary information includes the rank number, a grey bar that corresponds to the number of links on the page and some icons. There are two types of icons (see Figure 2). First, icons describe the file type of the retrieved link, this includes pdf, postscript, PowerPoint and Excel; the space is left blank if the linked document is html. Second, some icons are used to depict large quantities of various attributes. So for instance, if the document has a lot of images or lots of text then an icon appears, the threshold of this value can be set.

Another method that can aid the user in this strategy is to present a comparison of search results. Havre et al. [5] implemented the SPARKLER tool which compared multiple related queries. They believed that users would gain interesting insights from patterns of relevance values, brought about through the comparison of query results. Spoerri [18] implemented the MetaCrystal interface which visualized the intersections of search results from a query performed on different search engines. His intention was to cancel out the limitations of each search engine used by combining and comparing the results of the different search engines.

In recent work we [19] detail a summary view as part of our SES (search engine similarity) tool. The summary view displays a list of urls that appear in multiple searches and a circle icon depicting which search it exists within. We utilize this work as part of EvoBerry to display the list of urls that exist in multiple searches, and a number of rectangular icons denoting which search it exists in, and how highly it was ranked in that search. The Summary View is included in the ToolBox and can be seen in Figure 1. Websites which exist in multiple searches, and that have been generated from the same information need, have greater relevance.

3.2 Effective Visualization

Evolving searches generate large quantities of search results and need to be displayed in an way that can be easily browsed and understood. The results window is shown in Figure 2. Apart from visualizing the metadata and providing summary views for effective information seeking we include a focus+context technique that we named 'visual bracketing' [16]. The visual bracket allows the user to view some results in full detail (i.e. title, url, and text snipped) with adjacent results displayed in increasingly lower detail (see Figure 4); the further they are from the focused result the lower the detail.

Also, the results window includes a button panel from which the user can open webpages, bookmark pages (highlighting them in green), create a new (blank) page and zoom in and out to change the level of detail in the visual bracketing view. Shortcuts have also been implemented and users can access these functions through a right-click operation that initiates a popup menu.

3.3 Information recall

In EvoBerry, when a user clicks on a link in the results window to view a specific page, the page is loaded into the browser window (Figure 1) and a Visual History is started. EvoBerry implements various methods to enable effective information recall, including tabbed browser windows and Visual history bars.

Each webpage is loaded into a new tabbed pane (called a WebTab). Each WebTab contains basic web functionality of address bar, forward/back buttons, stop and refresh. Furthermore, the color of the WebTab corresponds with the color of the originating search result window. By using tab panes, the user can easily go back to a previous page.

At the bottom of the page we have developed a graphical history tool named the Visual History Bar (Figure 5). This provides a summary of the user's browsing operations from this location. Hasan et al. [4] acknowledged that diagrammatic summaries were important when navigating information spaces. All web-browsers provide *back* and *forward* navigation, and they mostly operate on a stack model where page information is lost when a user back-tracks and follows a new branch. However, our visual history bar utilizes a horizontal tree structure to enable the user to navigate and return to a previous page.

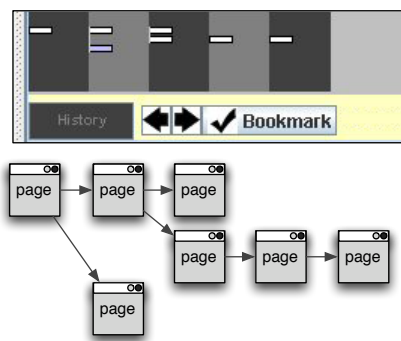


Figure 5: **The Visual History Bar.** On the top is a view of the bar, and on the bottom is diagram translating the diagram into the users real-world browsing progress.

A block is added every time a webpage is viewed; blocks, which are viewed successively, will appear at the same level horizontally. Blocks stacked on top of each and linked vertically by a yellow line represent links from the

same webpage. Finally, when the user hovers over a block the page title pops-up, and the page can be loaded by subsequently clicking a block.

3.4 Information management

The system contains different forms of management in different places. Each of the windows (browser, search result, and ToolBox) has their own management strategy, because they form different parts of the exploration.

Each results window has a session bar, which automatically records the users activity associated with that results window. This allows the user to load web pages, bookmark some pages, and return to them at any stage. Icons are added to the session bar representing search results, opened pages and bookmarked items. Right clicking on a session icon, displays a popup box, where the user can choose to reopen, copy or remove this item from the session. Clicking once on any session icon will highlight it and bring it into focus in the main view, and double-clicking on it will open the corresponding webpage. The user can also filter these icons, which only displays icons of a particular category, and they can be scrolled when there are many activities in the session. These icons display various attributes of the link that they represent. For example, a bookmark icon displays the rank number, or details of the types of link (see section 3.1).

The ToolBox contains three tools, the comparison tool, the dropbox and the notepad. The Comparison view was detailed in section 3.1. The Dropbox acts as a store for search results or pieces of information that the user wishes to actively save. Various types of information can be stored in the dropbox, and can be added to through an assortment of ways: (1) dragging a search result from the results window onto the dropbox stores all of its information as an item in the dropbox list, (2) by right clicking and specifying which piece of information to send (title, url, description) or (3) by typing text into the notepad tool and submitting it to the dropbox. Additionally, the user can write notes into the note section.

Finally, the Visual History Bar aids the user to manage where they have visited and return to previously viewed pages; its operation was detailed above (section 3.3).

4 Evaluation

To evaluate the EvoBerry environment we conducted an experiment comparing how users perform evolving searches using EvoBerry in comparison Microsoft's Internet ExplorerTM and search results gained from Yahoo's search engine. This pair was chosen because EvoBerry both uses IE to display and render the pages, and collects the results through Yahoo's search engine. Twenty test subjects were chosen for the experiment; each of them were experienced web users (i.e., people who use search engines

more than a couple of times per day). A within-subjects design was chosen for the experiment, and to minimize the impact of any learning effects the test-subjects were split into two groups, either doing EvoBerry first and then IE/Yahoo, and vice versa.

After an initial training session and system familiarization, the subjects were given two scenarios to complete, one for each interface. The basic premise was to force the subject to consult different data sources and to browse different locations using more than one criterion, and then to recall information they have viewed and to apply that knowledge to a new problem. Each scenario consisted of three questions based around the theme of planning a holiday. For example, one set of questions was: (1) find three countries that contain the 'big five' safari animals, (2) (assuming one of the countries is South Africa) find three National Parks in South Africa that contain the 'big five' safari animals, (3) for some reason you are unable to visit South Africa, hence find a country that contains the 'big five' that borders South Africa.

The evaluation data was gathered in four ways. First, EvoBerry automatically logs every user operation, including any button clicks, search terms used and pages viewed. Second, an 'answer sheet' program was developed in Java that sequentially asked the questions, stored and timestamped their answers. Third, the subjects filled in a post-experiment questionnaire. Fourth, the whole session was screen-captured. Two videos were recorded for each session (one each for EvoBerry and IE) with each video averaging 600-800 megabytes, in total over 30 gigabytes of video data was obtained.

4.1 Results & Analysis

The results received were positive, but complex in nature. It is clear that not all of our goals were fulfilled, but the results did not deviate far from the expected outcome, and in fact in some aspects, EvoBerry exceeded our expectations. A preliminary examination of the results, in comparison with the experiment hypotheses, indicate the following:

- Test subjects on average completed the search scenarios faster in the standalone Internet Explorer (IE) in comparison with EvoBerry.
- Test subjects were equally accurate in both the IE and EvoBerry
- Test subjects on average performed less searches in EvoBerry than in IE alone.
- Test subjects who used the comparison tool in EvoBerry were slightly faster in their tasks, performed slightly less searches overall but with the same accuracy.

The fact that test subjects were faster in completing the search scenario with IE alone was not surprising. It is certainly difficult to evaluate a novel interface in comparison with a well known and well used interface. But, in spite of the users' relative inexperience with the tool they still achieved scores the EvoBerry environment that equaled (and sometimes surpassed) their scores when using IE. This is a significant result, considering that the users would be less experienced with EvoBerry. In fact, the post-questionnaire supports this. Test subjects were innately more familiar with IE's style of web browsing, having been exposed to it for much longer, and as a result did not use the EvoBerry tools to their full potential. However, subjects who used the recall tools extensively (and used more than one tool) showed a gain in accuracy and always completed the recall question correctly.

In the questionnaire of all the EvoBerry tools the comparison tool was the most used and had many positive comments. Over half of the test subjects commented on the usefulness of the tool. One test subject said that the summary tool changed their 'way of searching' enabling them to drill down and discover pertinent results more quickly. Indeed, the video data confirms that multiple users relied heavily on the comparison tool: first generating searches and then consulting the comparison tool (to find any search intersections) before viewing the results. However this search behavior does result in the user performing large numbers of searches, which can fill the screen.

The 'result frames' window brought mixed responses. Some users liked that they separated search results from web pages, while others thought they took too much screen space. Finally, the history tool was also praised by those who used it. Saying that it was easy to locate pages and that it did not lose url information if the user chose another branch of the history tree.

5 Conclusion

We have performed a design analysis for evolving searches and have developed and presented an integrated environment (EvoBerry) to support the user in their exploration and we have performed an in-depth analysis of this environment.

Users definitely liked the environment and understood the reason for the tools. Test subjects were keen on the coordinated views and comparison tool although, as mentioned above, subjects found that it was hard to change their current searching strategies. Our experiments showed noticeable increases in performance (although not considered significant by the T-test) when utilizing the comparison tool. Furthermore, EvoBerry included various information recall techniques and those who used these tools

found them beneficial. Overall EvoBerry provides an effective environment to manage search-result sessions and provides an environment that supports the user in their evolving searches. Finally, the experience has certainly demonstrated that users need to be more familiar with the tool to become effective users: a challenge that any developer has to overcome with the development of any new tool or technique.

There are a number of improvements that can be made and further work that can be achieved. First, we are looking at ways in which the presented information and the user-interface can be simplified to make it easier and quicker for a user to find pertinent results. Second, we are investigating ways in which the features of the environment can be more closely integrated with the browser and also take up less screen space.

References

- [1] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.
- [2] J. Cugini, S. Laskowski, and M. Sebrechts. Design of 3D visualization of search results: Evolution and evaluation. In *Visual Data Exploration and Analysis*, pages 198–210. IST/SPIE, 2000.
- [3] G. W. Furnas. *The FISHEYE view: a new look at structured files*, pages 312–330. Morgan Kaufmann Publishers Inc., San Francisco, USA, 1999.
- [4] M. Hasan, D. Vista, and A. Mendelzon. Visual web surfing with Hy+. In *CASCON*, pages 28–38. IBM Press, 1995.
- [5] S. Havre, E. Hetzler, K. Perrine, E. Jurrus, and N. Miller. Interactive visualization of multiple query results. In *IEEE Symposium on Information Visualization 2001*, pages 105–112, Washington, DC, USA, 2001. IEEE Computer Society.
- [6] M. A. Hearst. *User Interfaces and Visualization*. Addison Wesley Longman, 1999.
- [7] T. Hewett, M. Czerwinski, M. Terry, J. Nunamaker, L. Candy, B. Kules, and E. Sylvan, editors. *Creativity Support Tools*. National Science Foundation, September 2005.
- [8] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, 2000.
- [9] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated. In *CHI '91: Human factors in computing systems*, pages 173–176, New York, USA, 1991. ACM Press.
- [10] T. M. Mann. Visualization of WWW-search results. In *DEXA Workshop*, pages 264–268, 1999.
- [11] G. Marchionini. Interfaces for end-user information seeking. *Journal of the American Society for Information Science*, 43(2):156–163, January 1999.
- [12] S. McCrickard and C. Kehoe. Visualizing search results using SQWID. In *6th World Wide Web Conference*, April 1997.
- [13] P. Pirolli and S. Card. Information foraging models of browsers for very large document spaces. In *AVI'98: Advanced Visual Interfaces*, pages 83–93. ACM Press, 1998.
- [14] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Human factors in computing systems*, pages 318–322, New York, USA, 1994. ACM Press.
- [15] J. C. Roberts, N. Boukhelifa, and P. Rodgers. Multiform Glyph Based Search Result Visualization. In *Proceedings Information Visualization 2002*, pages 549–554. IEEE, July 2002.
- [16] J. C. Roberts and E. Suvanaphen. Visual bracketing for web search result visualization. In *Proceedings Information Visualization 2003*, pages 264–269. IEEE Press, 2003.
- [17] G. G. Robertson and J. D. Mackinlay. The document lens. In *UIST '93: 6th ACM symposium on User interface software and technology*, pages 101–108, New York, USA, 1993. ACM Press.
- [18] A. Spoerri. MetaCrystal: visual interface for meta searching. In *CHI'04: extended abstracts on Human factors in computing systems*, pages 1558–1558, New York, USA, 2004. ACM Press.
- [19] E. Suvanaphen and J. C. Roberts. Explicit verses implicit: An analysis of a multiple search result visualization. In E. Banissi et al, editor, *Proceedings Information Visualization 2004*, pages 731–736. IEEE Press, July 2004.
- [20] A. Veerasamy and R. Heikes. Effectiveness of a graphical display of retrieval results. In *Research and development in information retrieval*, pages 236–245, New York, USA, 1997. ACM Press.