Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view

Patrick Baudisch
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
baudisch@microsoft.com

Bongshin Lee
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
bongshin@cs.umd.edu

Libby Hanna
Hanna Research & Consulting
1402 26th Avenue East
Seattle, WA 98112
libbyh@earthlink.net

ABSTRACT

Fishnet is a web browser that always displays web pages in their entirety, independent of their size. Fishnet accomplishes this by using a fisheye view, i.e. by showing a focus region at readable scale while spatially compressing page content above and below that region. Fishnet offers search term highlighting, and assures that those terms are readable by using "popouts". This allows users to visually scan search results within the entire page without scrolling.

The scope of this paper is twofold. First, we present fishnet as a novel way of viewing the results of highlighted search and we discuss the design space. Second, we present a user study that helps practitioners determine which visualization technique—fisheye view, overview, or regular linear view—to pick for which type of visual search scenario.

Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces. - User-centered design, Graphical user interfaces.

General Terms

Design, Human Factors.

Keywords

Fisheye view, web browser, search terms, popouts.

1. INTRODUCTION

Many web tasks require users to search inside a web page. A user may, for example, try to judge the relevance of a web page returned by a search engine based on the presence and grouping of relevant terms. Search functionality can help users get to relevant content faster whenever that content can be identified by using search terms. To accelerate such search tasks, web browser extensions such as the Google toolbar (toolbar.google.com) allow users to simultaneously highlight all occurrences of one or more search terms in the current page. By using different colors for each search term, highlights allow users to get a quick overview

Figure 1: (a) Displaying a long web page in a regular web browser results in page content being clipped. (b) Showing clipped content using an overview enhanced with popouts (similar to [27]).

of the relevant content of a page. The simultaneous perception of spatial properties, such as proximity and location of highlighted search terms, allows users to conduct searches that involve spatial relationships, such as "*rates* near(2) *cuts*" or "find the term *shop*, but only when located in the menu frame"—queries that would otherwise require learning more complex query syntax.

However, such search term highlighting works well only for pages that fit into the display window. Traditional web browsers clip long web pages, which hides all search results outside the current view (Figure 1a). This affects the user's ability to consider the page as a whole; to verify that a page does *not* contain a certain search term, users have to scroll through and inspect the entire page. To do more complex comparisons, users may even have to scroll back and forth. The ability to judge a page at a glance is lost.

In order to simplify search term scanning tasks in long web pages, Suh et al. proposed the Popout Prism which enhanced web browsers with an overview [27]. Figure 1b shows this technique, here implemented using the overview browser we used in our study. An additional column at the left of the browser window shows a miniature of the entire web page. An overlaid rectangle indicates which portion is currently viewed in the detail view on the right. To preserve the readability of search terms in the overview, Popout Prism overlays all occurrences of search terms in the overview with copies of that term that are enlarged to readable size (see also Figure 6a).

While overviews allow users to simultaneously see every occurrence of all the search terms in the page, they have their drawbacks. First, visual switching between overview and detail view requires users to reorient themselves. This wastes time [1], unless switching occurs only infrequently, in which case a continuous presence of the overview is unnecessary. Second, adding or removing an overview on the fly changes the page width available for the web page. This can cause the page to reflow, which may disorient users. And third, overviews frequently leave part of their screen space unused, especially for short pages (Figure 1b).

Focus plus context visualizations, such as fisheye views [7] use a single view, which frees them from all of the above limitations. This makes them a potentially interesting alternative for highlighted search inside web pages. Fisheye distortion can impair spatial cognition tasks [1], but fortunately spatial information plays only a minor role in web page-related activities. Given that web page layout often changes when the user resizes the browser window or adjusts the font size, or as images load, the amount of spatial information that web designers can encode into a web page is rather limited.

In this paper we therefore investigate this question: for what scenarios does highlighted search benefit from fisheye web browser, when is a page better viewed in an overview browser, and when does the traditional linear view work best?

The remainder of this paper is divided into two sections. In the first part, we present *fishnet*, a fisheye web browser prototype we built for this user study. Since the combination of fisheye web browsing and highlighted search has not been researched thus far, we dedicate a significant amount of the paper to exploring the design space. In the second part of the paper, we present the results of our user study in which we compared fisheye view, overview, and linear view web browsers, all offering highlighted search. We report both quantitative and qualitative measures, and discuss design implications for the display of web browsers supporting highlighted search.

2. FISHNET WALKTHROUGH

Figure 2 shows a screenshot of the fisheye web browser fishnet. Fishnet always displays web pages in their entirety, independent of their size (exception: preformatted pages may result in a *horizontal* scrollbar if shown in a narrow browser window). Fishnet accomplishes this by showing only a focus region at readable scale, while compressing page content above and below the focus region. These compressed context areas are rendered slightly darker to help users visually pick out the different areas.

Figure 3 shows how a web page is scrolled in fishnet. As the user scrolls the web page down, the focus area moves downwards, continuously moving content from the bottom context into the focus area and from the focus area into the top context.

Fishnet offers search and highlighting capabilities comparable to the Google Toolbar. Fishnet also preserves the readability of search terms at all times by overlaying them with popouts whenever off-focus (see also Figure 6b).

The functionality offered by fishnet is similar to the functionality offered by a corresponding overview solution. First, fishnet allows users to *read* part of the web page's content—that part currently located in the focus area. Second, fishnet allows users to visually *scan* the entire page for highlighted search results, without the need to scroll.

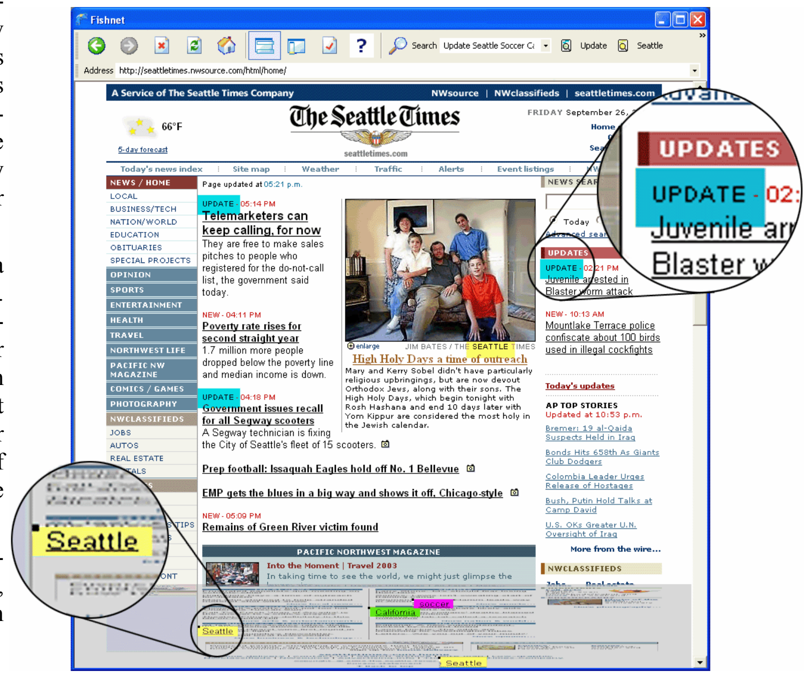

Figure 2: The fishnet web browser fits web pages of any length into the browser window by vertically compressing peripheral content. Color-coded popouts keep search terms ('Seattle', 'update', …) readable while they are in one of the context areas.

Fishnet has a different set of strengths and weaknesses than overview solutions. Strengths include the fact that fishnet always makes use of all available screen space. The context areas can be added on demand, i.e. as the page overflows the browser window size, without having to reflow the page. When traversing a page top to bottom, the focus area helps orientation by providing a point of reference for what has been looked at (above) and what has not yet been looked at (below). Fishnet's weaknesses include the fact that the aspect ratio of content in its context areas is affected, and that scrolling a piece of content into the focus area can lead to over- or undershooting [10]. Also, the vertical compression of content in the context areas can become significantly higher than in an overview browser.

In the remainder of this paper, we go over the related work, discuss design decisions behind fishnet, and present the results of a controlled experiment comparing fishnet, overview, and linear view with respect to four types of visual search tasks. We conclude with a discussion of the findings, design recommendations, & plans for future research.

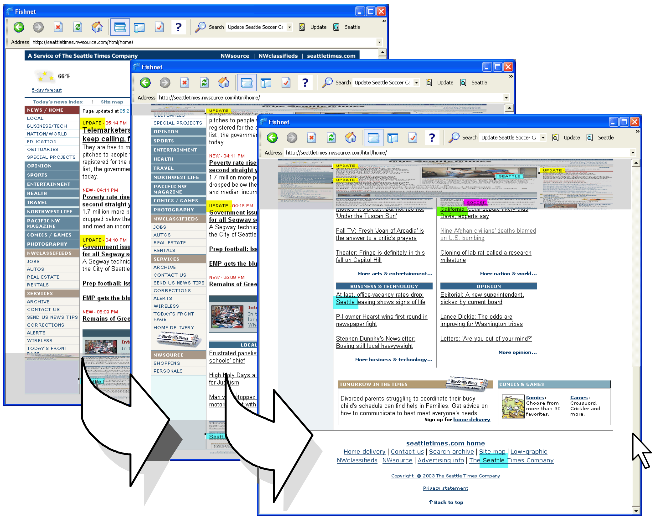

Figure 3: Scrolling a web page in fishnet moves the focus area down the page.

3. RELATED WORK

A substantial amount of research has been done to help users view and scan large documents on limited screen space. The most basic approach is to use zooming and panning to display required information *sequentially* (e.g., [14]). In order to reduce clutter, material may change its representation when zoomed (*semantic zooming*, e.g., [3]).

Overview plus detail visualizations save navigation effort by displaying a zoomed-in and a zoomed-out view simultaneously [25]. In addition to the navigation of large spatial documents [21], overviews have been used to help users traverse text documents, such as program code [6], more efficiently. In *Reader's Helper* the thumbnail scrollbar shows an overview of a page including highlighted annotation and topic scores [8]. *Popout Prism* [27] improves this concept for web search tasks by introducing more sophisticated highlighting for search terms called popouts (similar to *enhanced thumbnails* [27]).

Focus plus context visualizations [26] have also been applied to large documents, such as maps [16], [24], graphs [17], tables [22], calendars [4], and file systems [18] and were found appropriate for large steering tasks [11]. Various styles of fisheye views have been proposed for browsing text documents, including browsers for a single page [15], multiple pages laid out in 2D space (*Document Lens* [23]), or multiple documents arranged in a rearrangable tile structure (*Zoom Browser* [12]).

Only a few papers so far have compared the usability of these interfaces. Keahey and Marley compared a fisheye text browser with the UNIX text reader *less* [15]. They found that users performed faster with the fisheye when browsing structured text, but slower when browsing unstructured text. Hornbæk and Frøkjær compared a fisheye view with an overview and a linear view [13]. They found that participants read documents faster when using the fisheye interface than when using an overview, but at the expense of reduced comprehension of the browsed material.

The work presented in this paper is different from the prior work in that we examine the combination of fisheye views with search term highlighting/popouts, a technique that allows users to make active use of otherwise unreadable off-focus content. We present a quantitative evaluation of a fisheye view, an overview, and a linear view interface, all offering corresponding highlighting features. Note that this is also the first quantitative evaluation of popouts applied in an overview.

4. FISHNET DESIGN

Fisheye visualizations come in a wide variety of styles [11]. Fishnet is designed to support two main activities: *scanning* pages for search terms and *reading* page content.

To help users scan a page, fishnet always shows the page in its entirety and preserves the readability of search terms throughout the page by using popouts. This is a qualitative improvement over a clipped representation of a web page, as the fisheye representation allows users to verify at a glance not only the *presence* of a particular search term or search term combination, but also its *absence*.

To help users read a page, fishnet is designed to render as much page content as possible at readable size. In order to achieve this, Fishnet is optimized according to the following four objectives. One, render as much material as possible at full size. Two, when scaling down, use scaling techniques that preserve readability as much as possible. Three, avoid disrupting page elements. And four, make content fill all available space if that improves readability, rather than leaving space blank.

We now look at the individual design decisions we made in order to accomplish these design objectives.

4.1 Distortion geometry

There are several different methods that could be used to accomplish scaling of off-focus content. Reducing the size of each content element individually (e.g. [13]) preserves the aspect ratio of content and is space-filling, but reflows content. Since this may affect users' spatial memory and prevent users from being able to track a page element as the page is scrolled, we limited ourselves to methods that do not cause page reflow. Changing scale abruptly as the resolution boundary is crossed can make elements on the edge unreadable (Figure 4a). Scaling objects as a whole during the transition (Figure 4b) works for single line objects (e.g. [2]), and groups of objects (zoomscapes [9]), but fails for very large objects—a page containing a single table could never shrink. Central perspective (e.g. [18]) solves this by shrinking content continuously with increasing distance from the focus (Figure 4c).

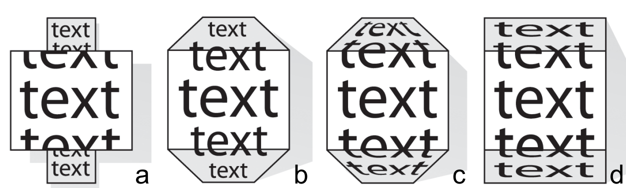

Figure 4: (a) Manhattan lens, (b) zoomscapes (c) central perspective, and (d) the parallel projection used by fishnet.

Fishnet uses parallel projection (Figure 4d). Off-focus content is compressed only vertically and all by the same factor. Compared to central perspective we found this model to have several benefits. First, it makes use of the entire screen space offered by a rectangular window. Second, it preserves horizontal alignment, which turned out to be particularly useful for understanding content arranged in tables. Third, it prevents the top and bottom of the page from becoming readability bottlenecks. And fourth, it improves readability by assuring constant line spacing. On the downside, parallel projection affects the aspect ratio of content more than other distortion styles.

In order to allow comparison with other fisheye models implemented in 3D [5], Figure 5 illustrates fishnet's geometry as a 3D model. When modeled in 3D, fishnet has a bridge-like geometry. The slanted parts represent the context areas. The 2D screen image is obtained by rendering the bridge from a viewpoint straight above. To prevent the trapezoid distortion of the context areas, parallel perspective is used, which can be thought of as observing the bridge from an "infinite" distance. The scene is illuminated using only ambient light, which makes the slanted parts appear darker (actually, fishnet reduces contrast of content in theses areas, similar to [27]). The higher the compression of the context areas, the more slanted the additions are and the darker they are rendered.

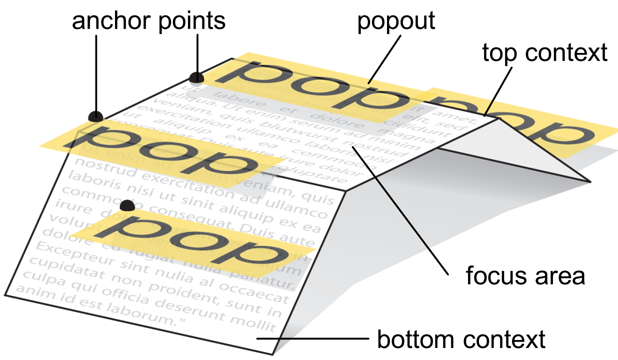

Figure 5: In 3D, Fishnet can be modeled as a bridge-like geometry. Popouts float on top of regular page content.

4.2 Scaling content

Fishnet always renders content in the focus area at 100% scale. Fishnet determines the scale factor of the context area as follows: If the page fits into the browser window, there is no context area and all content is rendered at 100%. For pages longer than the browser window, fishnet tries to fit the page into the browser window using bicubic filtering and scale factors between 50% and 25%. Fishnet avoids scale factors close to 100%, because any resampling reduces the readability of content [20], so that scaling a large portion of a page "a bit" creates a higher loss in readability than scaling a small portion by a higher factor. If scaling to 25% still exceeds the context size threshold, fishnet gives up on readability in the context areas and simply fits the context into a given size limit.

4.3 Highlighting and popouts

Fishnet implements highlighting by changing the background color around each appearance of a search term. Since the readability of a text element is generally guaranteed if rendered at 100% scale (otherwise the author of the page would have chosen a different size), fishnet never increases the size of search terms. Fishnet increases the saliency of search terms instead by increasing the size of the *highlight*. The extended highlight is rendered behind text content to avoid occluding content. By default, the extension adds 8 pixels in all four directions (Figure 6b).

Figure 6: (a) popout [27] vs. (b) extended highlight

Fishnet provides search terms with *popouts* whenever they appear in context areas. By default, popouts use 100% scale, but they can be configured to be vertically compressed to reduce the occlusion of page content and other popouts.

Fishnet anchors popouts to the page at the point closest to the focus area (see the two highlighted anchors in Figure 5). Any other type of anchoring would cause popouts to get disrupted during the traversal, because the popout fragment located over the focus moves at a different speed than the fragment over the context area.

4.4 Mouse interactions & keyboard shortcuts

Fishnet supports the same navigation controls as other web browsers, such as MS Internet Explorer™. This includes scrollbar, mouse wheel, and cursor key navigation. In addition, clicking any location in the context area scrolls that part into focus ("click navigation"). Slow-in slow-out animation is used to help users keep track of those locations. If a popout is clicked, a red frame is added to that popout throughout the animation, helping users keep track of that term. Fishnet search is operated exactly like the Google toolbar, including buttons that allow scrolling to the next occurrence of the respective search term. Additional buttons in the toolbar allow users to activate and deactivate the fisheye mode, so fishnet doubles as a linear web browser (and also as an overview browser, see user study section).

5. IMPLEMENTATION

Fishnet is a stand-alone executable, written in C#.NET. Fishnet uses the Microsoft WebBrowser Control to parse and render web pages (http://msdn.microsoft.com). The control fills the entire fishnet window; the context areas are rendered on top of it.

Fishnet loads pages as follows: Since a web page can be captured only after the web page is fully downloaded, fishnet moves the WebBrowser Control temporarily off-screen to prevent user interaction. There, Fishnet highlights all search terms by changing the background color and adds highlight extensions by positioning colored tables under the highlighted search terms using the SPAN html tag. If the page does not fit in the WebBrowser Control, Fishnet scrolls the view horizontally and vertically, taking screenshots each time. Then it stitches all screenshots into one image. Fishnet scales the image down using bicubic filtering, and tints it using brightness and contrast filters. To improve scrolling performance, fishnet pre-renders popouts into the context images, pre-scaled if necessary.

On repaint, the WebBrowser Control renders itself; then fishnet adds the context areas by copying the respective fragments from the context image over the browser control. On a 2 GHz machine fishnet scrolls a 1024x768pixel browser window at around 20 frames/sec.

6. USER STUDY

The purpose of our user study was to determine which of the three visualization techniques (fisheye view, overview, and linear view, each with search term highlighting) works best for which class of scenarios. Prior work suggests that the performance of all three techniques depends strongly on the properties of the viewed page, such as page structure (e.g. [15]). The goal of our study was to help practitioners choose which technique is appropriate for a given scenario. In particular, we focused on levels of understanding of the spatial arrangement of search terms in the page. We investigated how much these task requirements impact the performance of fisheye, overview, and linear view.

6.1 Interfaces

There were three interfaces in the experiment. The *fisheye interface* was fishnet in the configuration described in this paper. The *overview interface* was implemented using fishnet as well (Figure 7b, c). This mode was identical to Popout Prism, but to bring it up to par with the fisheye interface, the detail view of the overview interface used extended highlights instead of popouts; this eliminated the need for users to manage popouts visibility. Also the *linear interface* was implemented using the fishnet browser, in this case with overview and fisheye turned off (Figure 7d).

All three interfaces occupied the same amount of screen space. The linear interface offered an 860x800 pixel content area, which allowed viewing the web pages used in the study (preformatted layouts of about 800 pixel widths) without leaving unused space. The fisheye and overview interfaces obtained space for their visualization features by trading in a certain percentage of detail view space (unlike [13]). The 287 pixels wide overview of the overview interface was added on the left; the height of this browser was shortened to 600 pixels to keep screen space compatible to the other conditions (overall 1147 x 600 pixels). The resulting difference in aspect ratio (see Figure 7) was unavoidable, as using identical aspect ratios would either have created a horizontal scrollbar for the overview interface or unused space for the other interfaces. Differences in aspect ratio, however, seem less important today, as PC displays can be rotated in software.

All three interfaces offered the same search functionality, and the same navigation shortcuts, e.g., clicking on a search term button in the toolbar allowed jumping to the next occurrence of that term. In addition, the fisheye interface allowed click navigation, and the overview interface allowed clicking and dragging the overview. Clicking or dragging on a popout in the overview created a red rectangle around that term in both overview and detail view.

6.2 Tasks and procedure

There were four tasks (Figure 7). All tasks required participants to analyze a web page and then use one or several checkboxes and a submit button. In all tasks, the browser's search term box was initialized with a set of search terms, so that relevant terms in the page were already highlighted. These terms are highlighted in the following description. The search term box, as well as links and all other browser toolbar functions were disabled, so participants were not able to change the search terms.

All web pages were created from pages downloaded from public websites. Variations for individual trials were made by disassembling the original page into components, such as table cells or paragraphs, and reassembling the page from a randomly selected subset of these components. Web pages filled two to four pages when displayed in the linear interface.

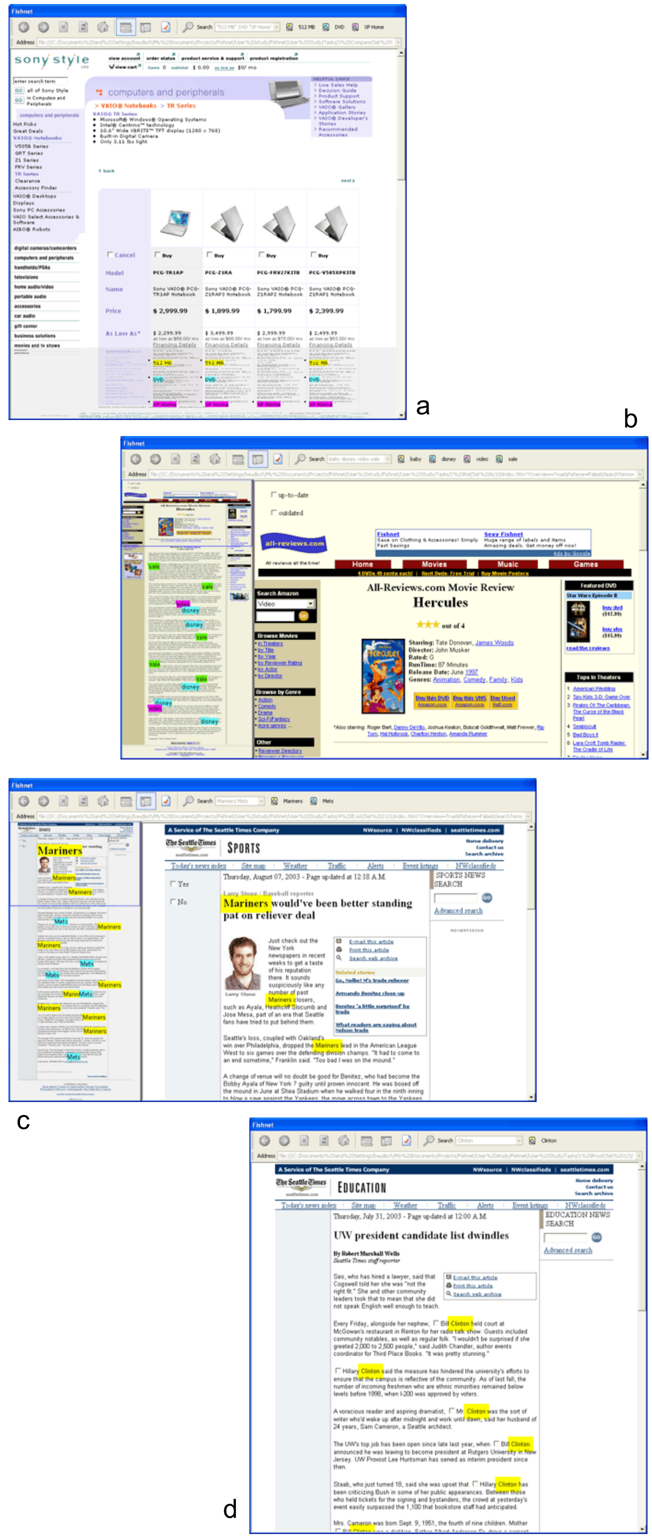

Figure 7: The tasks (a) product choice, (b) outdated, (c) co-occurrence, and (d) analyze, shown in the fisheye, overview, overview, and linear interface.

Outdated task (http://all-reviews.com, Figure 7b)*:* Each page contained several paragraphs of information about gifts. The participant's task was to check whether this page contained all four search terms baby, disney, video, and sale. If so, this would indicate that the page was up-to-date and participants would click the corresponding checkbox at the top of the page. If not, they would click the "outdated" checkbox.

Product choice task: (http://sonystyle.com, Figure 7a). Each page displayed a table with four columns containing feature lists for notebook computers. The participant's task was to click the checkbox in the column of the cheapest notebook that offered 512 MB memory, a DVD player, and Windows XP Home Edition, or to click the "cancel" checkbox if none of the laptops offered all 3 features.

Co-occurrence task (http://seattletimes.com, Figure 7c)*:* Each page contained several paragraphs with information about baseball teams. The participant's task was to check whether the page contained any paragraphs that contained both search terms Mariners and Mets. If so, they would click the "Yes" checkbox at the top of the page. If not, they would click the "No" checkbox.

Analysis task (http://seattletimes.com, Figure 7d)*:* Each page contained an article about Hillary and Bill Clinton. The participant's task was to check how often Hillary Clinton was mentioned, e.g. Mrs. Clinton, or Hillary R. Clinton, etc. The term Clinton was highlighted, but the participant had to check the individual matches to see whom they referred to. Participants counted occurrences by clicking checkboxes located next to Mrs. Clinton's name.

Using a within subject design, we had each participant complete all 4 tasks on all 3 interfaces, with 4 training and 10 timed pages for each task. Performance measures were task completion time and error rate. Task, interface and page order were counterbalanced across participants. After completing each task x interface combination, participants answered questions about their subjective satisfaction; they completed another overall satisfaction questionnaire at the end of the study. All participants completed the study in less than 60 minutes.

6.3 Participants

Thirteen participants (6 women) were recruited from <removed for review>. They were between 22 and 49 years old. All participants reported spending a minimum of 10 hours a week using the Internet and demonstrated an "intermediate or above" knowledge of Internet terminology.

6.4 Hypotheses

Our main hypothesis was that the success of each of the three interfaces would be determined mainly by the requirements of the task. Table 1 shows how the four tasks can be classified according to two variables. Only the tasks in the right column require participants to discriminate how information is arranged in *columns*; only the tasks in the bottom row require participants to discriminate how information is arranged in *rows*. The table cells then summarize our individual hypotheses.

For the outdated task, participants needed to check the existence of search terms—spatial arrangement played no role. This should give the fisheye and overview interfaces a benefit over the linear interface, as they both eliminate the need to scroll.

For all other tasks, spatial arrangement does play a role. In the product choice task, seeing in which *column* a search term is located is key; seeing what *row* a search term is in, on the other hand, is not important, as the order of product features in the table is immaterial. These requirements should favor the fisheye interface over the overview, as the fisheye offers higher resolution in horizontal direction.

The requirements for the co-occurrence task are diametrically opposed to the product choice task. In this task, seeing in which paragraph a search term is located is key. The vertical resolution of the overview interface should still support this, while the strong vertical compression of the fisheye's context areas should not provide enough resolution to judge whether two elements are located in the same paragraph.

The analysis task, finally, forms the baseline. It requires participants to read detail information, making both overview and fisheye view virtually useless. In this task, the linear interface should perform slightly better as it offers the largest reading area.

Table 1: Our hypotheses about which interface should perform best for each of the four tasks

	distinguishing columns immaterial	…necessary
distinguishing rows immaterial	outdated task → overview&fisheye	product choice task →favors fisheye
…necessary	co-occurrence task →favors overview	analysis task →favors linear view

6.5 Results

Task completion time: We evaluated differences between the interfaces and tasks using a repeated-measures ANOVA on a logarithmic transformation of each participant's average time across trials. Time was the dependent variable, and task and interface were the independent variables (Figure 8). All main effects and the interaction between task and interface were significant, $p<.001$.

In order to examine the interaction between task and interface, we performed repeated-measures ANOVAs separately for each task. These analyses included order as an independent variable to check for order effects. The ANOVAs also allowed for post-hoc tests to determine which interfaces were statistically different from one another within each task (using the Bonferroni correction for multiple comparisons). There were no main effects for order of interface, so order is not discussed further here.

In the outdated task, the main effect for interface was significant at $F(2,14)=29.74$, $p<.001$. Pairwise comparisons of interfaces showed that the overview and the fisheye interfaces were significantly faster than the linear interface, $p<.001$, as expected. The overview and the fisheye interface were virtually identical to one another. When using overview or fisheye interface, participants verified the colors of the popouts on the page without navigating, while the linear interface required scrolling.

In the product choice task, the main effect for interface was significant at $F(2,14)=47.38$, $p<.001$. The fisheye interface was roughly twice as fast as the other two interfaces. The overview and linear interfaces were not different from one another. When using the fisheye interface, participants quickly scanned the columns to determine the correct answer. In the overview interface, where content was also horizontally compressed, participants tended to scroll or click the popouts to "check" the information in the overview pane, especially when the popouts overlapped columns in the overview pane.

In the co-occurrence task, the main effect for interface was significant at F(2,14)=6.49, p<.01. The linear interface was significantly faster than the fisheye interface, but not from the overview interface. The fisheye and the overview interfaces were not different from one another. When using the fisheye interface in this task, participants weren't sure if popouts that were close to one another or overlapping in the context areas were located in the same paragraph. They scrolled or navigated down the page to check and re-check individual paragraphs.

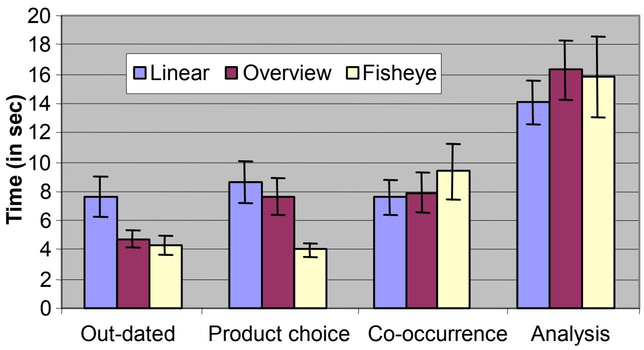

Figure 8: Average task completion times in seconds (error bars indicate 95% confidence intervals around averages).

In the analysis task, the main effect for interface was *not significant*, F(2,14)=5.37, p<.02 (using a corrected criterion of p<.0125 because of the multiple ANOVAs). The post-hoc comparison found only one statistical trend, for the linear interface to be different from the overview interface, p=.07. The lack of stronger differences in this task was expected due to the required navigation in all three interfaces to read and/or click by all popouts in the page.

Error rates: A Chi-Square analysis of error rates within interfaces and tasks found only one significant result, within the co-occurrence task (Chi-Square=7.147, p<.03). The error rate for the fisheye interface was 6%, compared to 2% in the linear interface and 0% in the overview interface. This is probably due to the confusion over overlapping popouts in the context areas for the fisheye interface. Overall, error rates were generally highest for the analysis task (7%–9%) and lowest for the outdated task (0%–1%).

Subjective preference: At the end of the sessions, participants ranked the interfaces in order of their preference. Ten out of thirteen participants preferred the overview interface over both the others, with the remaining three preferring the fisheye interface.

These results are illustrated in part by the subjective ratings participants provided after using each interface + task combination. Figure 9 shows the averages for the rating on "Using this interface for this task, it was easy to complete the task." (7-point Likert scale, with 1 = completely disagree and 7 = completely agree). Two-factor ANOVAs on each rating found significant main effects for all interface and interface by task interactions, p<.001. The ratings generally reflect task completion times. The only exception was the overview interface, which received relatively positive ratings on all tasks, including the co-occurrence task. The other two questions in the questionnaire ("…I felt I had a good sense of the page as a whole" and "…it was easy to locate all the highlighted words on the page") showed the same trends as Figure 9, so we omit reporting them in detail here.

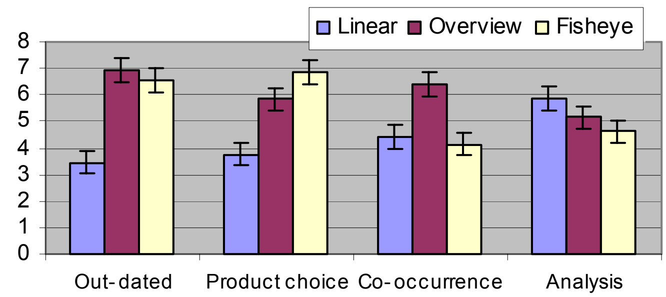

Figure 9: Average *ease of task completion* ratings for each interface and task (error bars indicate 95% confidence intervals around averages)

Comments about the overview interface included "allows quick glance searches" and "best sense as a whole." Feedback about the fisheye interface included "keeps everything in perspective a little better" and "very easy to see the entire page", but also negative comments, such as "couldn't read the text" and "lost track of where I was when I scrolled". The reaction to the linear interface can be summed up in one comment: "All that scrolling—need I say more!"

7. DISCUSSION

Our study suggests that the success of each of the three interfaces depends strongly on the task. The performance of the fisheye interface, in particular, was determined by the organization of content in rows. In cases where task completion did not require users to discriminate rows, the fisheye interface clearly outperformed both competing interfaces—resulting in a drop of 55% in task completion time (outdated and product choice tasks). On the other hand, if rows did play a role the performance of the fisheye interface fell back to the performance of the linear interface. The fisheye interface then actually increased task completion time by 18%, as the fisheye interface had wasted some of its space for the now useless context area.

Overall, the speed gained by the fisheye interface in the row-independent tasks was bigger than the speed lost in the row-dependant tasks, so with all four tasks taken together, the fisheye interface ended up being the fastest of the three examined interfaces (an overall average of 8.37 for the fisheye interface compared to 9.13 for overview and 9.48 for linear). Subjective satisfaction, however, was not fully in line with the performance results. While all participants more or less liked the overview interface, the fisheye view—an interface style none of the participants were familiar with—polarized participants; three participants ranked it first but six ranked it last. A long term study is required to investigate whether users' subjective preference may reverse itself as users gain more experience with this still fairly uncommon visualization style.

7.1 Design recommendations

Our findings suggest that the combination of highlighted search with any of the three evaluated interface styles results in a useful interface, but each for a different type of scenario. Based on this observation, we offer the following design recommendations.

1. Consider offering both fisheye and overview. Even though they are interchangeable for some tasks (e.g., in the outdated task), some tasks that require spatial discrimination favor the fisheye view, others the overview.

2. Allow users to bring up overview or fisheye view on demand, e.g., by pressing a shortcut. To not disrupt the user's spatial memory, this should be done without causing the page to reflow. The fisheye view does this automatically, as the context areas are only overlaid onto top and bottom of the page. In the overview case, consider compressing the detail view horizontally to make space for the overview.

3. Consider using context areas of minimal size. As the study showed, fisheye views lose the ability to discriminate row/vertical information already for page lengths around three pages. The remaining functionality, i.e., an indication of the presence and the horizontal position of search term, can be achieved with *very* thin context areas that basically only serve to anchor popouts (see *city lights* [19]).

7.2 Future research

The study presented in this paper focused on the effect of spatial variables on performance. In future research, we plan to investigate the impact of page length/browser windows size and the amount of in-page navigation required by the task.

8. ACKNOWLEDGMENTS

Sara Su, Jeremy Thorpe, and Duke Hutchings for participating in the pilot study. Sumit Basu for his comments on the study design.

9. REFERENCES

[1] Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proc. CHI'02,* 259–266.

[2] Bederson, B. Fisheye menus. In *Proc.UIST'00,* 217–225.

[3] Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *J. of Visual Languages and Computing,* 7, 3–31, 1996

[4] Bederson, B., Clamage, A., Czerwinski, M., & Robertson, G. DateLens A Fisheye Calendar Interface for PDAs. *ACM TOCHI,* (in press).

[5] Carpendale, M.S.T., Montagnese, C. A. Framework for unifying presentation space. In *Proc. UIST 2001,* 61–70.

[6] Eick, S. G., Steffen, J.L., and Sumner, E.E. (1992) Seesoft—A Tool for Visualizing Line Orientated Software Statistics. *IEEE Transactions on Software Engineering,* 18(11-Nov), 957-968.

[7] Furnas, G. Generalized Fisheye Views. *Proc.CHI'86,* 16-23.

[8] Graham, J. The Reader's Helper: A Personalized Document Reading Environment. *Proc. CHI'99,* 481–489.

[9] Guimbretière, F. Stone, M. and Winograd, T. Fluid Interaction with High-resolution Wall-size Displays. *Proc. UIST'01,* 21–30.

[10] Gutwin, C. Improving Focus Targeting in Interactive Fisheye Views. *Proc. CHI'02,* 267–274.

[11] Gutwin, C. Fisheye Views are Good for Large Steering Tasks. In *Proc CHI'03*, 201–208.

[12] Holmquist, L. E. and Ahlberg, C. The Zoom Browser: Presenting a Focus + Context View of World Wide Web Documents. *Technical Report,* SSKKII Gothenburg University, 1997.

[13] Hornbæk, K. and Frøkjær, E. Reading of Electronic Documents: the Usability of Linear, Fisheye, and Overview + Detail Interfaces. *Proc. CHI 2001,* 293–300.

[14] Igarashi, T. and Hinckley, K. Speed-dependent Automatic Zooming for Browsing Large Documents, *Proc. UIST 2000*, 139–148.

[15] Keahey, T. A. and Marley, J. Viewing Text With Non-Linear Magnification: An Experimental Study, *Technical Report 459,* Department of CS, Indiana Univ., 1996.

[16] Keahey, T. A. The Generalized Detail-In-Context Problem, *Proc. IEEE Symposium on Information Visualization'98,* 44–51, 1998.

[17] Lamping, J., Rao R., and Pirolli, P. A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proc. CHI 1995,* 401–408.

[18] Mackinlay, J. D., Robertson, G., and Card, S. The Perspective Wall: Detail and context smoothly integrated. *Proc CHI 1991,* 173–179.

[19] Mackinlay, J., Good, L., Zellweger, P., Stefik, M., and Baudisch, P. City Lights: Contextual Views in Minimal Space. *CHI 2003 Extended Abstracts,* pp. 838–839.

[20] Nyquist, N. Certain topics in telegraph transmission theory. *Trans. AIEE,* vol. 47, 617–644, Apr. 1928.

[21] Plaisant, C., Carr, D., & Shneiderman, B. Image-Browser Taxonomy and Guidelines for Designers, *IEEE Software,* 12(2):21–32, 1995.

[22] Rao, R., and Card, S.K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. *Proc. CHI 1994.* 318–322.

[23] Robertson, G. and Mackinlay, J. The Document Lens. *Proc. UIST 1993.* 101–108.

[24] Sarkar, M., Snibbe, S., Tversky, O.J., and Reiss, S.P. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens, *Proc. UIST'93,* 81–91.

[25] Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction. Third edition.* Reading MA: Addison-Wesley, 1998.

[26] Spence, R. and Apperley, M. Data base navigation: An office environment for the professional. *Behaviour and Information Technology,* 1(l), 43–54, 1982.

[27] Suh, B., Woodruff, A., Rosenholtz, R., and Glass, A. Popout Prism: Adding Perceptual Principles to Overview + Detail Document Interfaces. *CHI'02,* 251–258.

[28] Woodruff, A., Faulring, A., Rosenholtz, R. Morrison, J. and Pirolli, P. Using thumbnails to search the Web. *Proc. CHI 2001,* 198–205.