Transient Visualizations

Mikkel Rønne Jakobsen
Department of Computer Science
University of Copenhagen
Universitetsparken 1, DK-2100, Denmark
+45 35321451
mikkelrj@diku.dk

Kasper Hornbæk
Department of Computer Science
University of Copenhagen
Universitetsparken 1, DK-2100, Denmark
+45 35321425
kash@diku.dk

ABSTRACT
Information visualizations often make permanent changes to the user interface with the aim of supporting specific tasks. However, a permanent visualization cannot support the variety of tasks found in realistic work settings equally well. We explore interaction techniques that transiently visualize information near the user's focus of attention. Transient visualizations support specific contexts of use without permanently changing the user interface, and aim to seamlessly integrate with existing tools and to decrease distraction. Examples of transient visualizations for document search, map zoom-outs, fisheye views of source code, and thesaurus access are presented. We provide an initial validation of transient visualizations by comparing a transient overview for maps to a permanent visualization. Among 20 users of these visualizations, all but four preferred the transient visualization. However, differences in time and error rates were insignificant. On this background, we discuss the potential of transient visualizations and future directions.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces, I.3.6 [**Methodology and Techniques**]: Interaction Techniques

General Terms
Design, Human Factors

Keywords
Interaction techniques, visualization, transient, lightweight, fluid, overview+detail, fisheye

1. INTRODUCTION
Many information visualizations make permanent changes to the way the visual structure of information appears in the user interface. Different mechanisms are used toward this change, such as transforming the visual structure, adding features to the visual structure, and using multiple views. For example, Fishnet [1] permanently applies a bifocal display transformation and adds search-term popouts to the visual structure of a web page, and Popout Prism [15] permanently adds a zoomed-out overview to the detail view of a web page.

Designing permanent visualizations that are suitable for realistic work environments is complicated by the diversity of tasks that need to be supported. For example, consider a fisheye view of source code that presents context information relevant to the current focus. Such as view may support navigation and understanding, but the same fisheye view is inappropriate for writing and editing code because programmers want a large view of source code for those tasks [11]. Based on the observation that a particular design of a permanent visualization may be suitable only in some scenarios, Baudisch et al. [1] recommended that users should be allowed to bring up visualizations on demand, for example by using a keyboard shortcut.

We discuss transient visualizations, interaction techniques for transient use of information visualizations close to the user's focus of attention. Many user interfaces successfully employ techniques that provide users with transient information in the context of their focus of attention, including tool tips and context menus. Also, the HCI literature presents numerous techniques that involve transiency, lightweight interaction, and visualization [e.g., 4,9,10]. However, we are unaware of any attempts at generalizing about using information visuali-zations transiently. Therefore, the general benefits of transient visualizations and the factors that advance and restrict their use are unclear. In this paper, we present examples to probe potential benefits of transient visualizations, and report an initial validation of one instance of a transient visualization.

Contributions of our work are (a) to direct researchers' awareness toward transient uses of information visualizations that may help avoid problems inherent in the design of permanent visualization interfaces, (b) to provide a basis for practitioners to consider how transient visualizations may be utilized in the work practices they seek to support, and (c) to present encouraging initial data about the usability of transient visualizations.

2. CHARACTERISTICS OF TRANSIENT VISUALIZATIONS
Transient visualizations have four characteristics:

- Immediacy; to bring the user into direct and instant involvement with the information representation.
- Transiency; information is only displayed tempo-rarily, and is easily dismissed, which means that no display space is used permanently.
- Closeness; the information is shown close to the region of focus in the display (e.g., cursor or caret), resulting in fast access to the information because of minimized sensory-motor efforts of the user.
- Contextuality; the information is related to the user's current focus of attention, for example by adding context for interpreting the information in focus.

OzCHI 2007 Proceedings, ISBN 978-1-59593-872-5

We contrast transient visualizations with permanent information visualization interfaces, such as overview+detail interfaces where permanent display space is allocated to an overview window [15]. First, designers are challenged with deciding what information is needed in various contexts of use and fitting the information into the limited display space of a permanent visualization. In contrast, using transient visualizations to facilitate infrequent and unpredictable contexts of use, the original permanent view can be dedicated to information used in frequent contexts of use.

Second, adopting permanent visualizations to improve an existing tool may break established uses of the tool. However, the means of invoking and interacting with transient visualizations can be tailored to particular contexts of use, thus supplementing established interaction habits.

Finally, rich and dynamic views in permanent visualizations may visually disorient and annoy the user. In contrast, using transient visualizations that appear only temporarily and under the user's control helps prevent visually complex and disorienting interfaces.

3. EXAMPLE APPLICATIONS

To provide concrete arguments for the potential of the idea of transient visualizations, we present sketches of transient visualizations that support tasks in three different domains, and describe a prototype of a transient visualization in a programming environment.

3.1 Searching in Documents

In conventional web browsers, 'Find' automatically jumps to the first instance of words as they are typed. However, scrolling between found instances may disorient the user [15]. Recent studies have shown overview+detail visualizations [15] and bifocal displays [1] to be efficient and preferred by users for searching in documents. Among the advantages experienced by participants using an overview+detail interface, Suh et al. [15] report that the interface gives "a sense of context, density, and the ability to see all occurrences of a keyword at once" and provides orientation support for navigating in the document.

In the design mock-up in Figure 1, we show a transient visualization to support in-document search; our approach extends a conventional browser window by calling up a thumbnail overview when the user invokes the 'Find'-bar. As the user begins to type keywords, instances of the words are highlighted in the overview. The user can move between highlighted words using the keyboard, or drag the field-of-view window using the mouse. Behind the overview, the original view scrolls the document accordingly, visually coupling the overview to the original view. Finally, the overview can be dismissed to scroll back to the original location in the document by suspending the 'Find' action (e.g., with the Escape-key).

We believe that our suggested design provides the same support for in-document search as a permanent overview by giving an overall sense of the location, density and co-occurrences of keywords. Additionally, in contrast to a permanent overview, (1) the overview does not compete for permanent display space, and (2) fluid keyboard interaction allows the user to complete their task without having to switch to mouse. In reading tasks, Hornbæk and Frøkjær [8] found overviews to support navigation and help to get a structural overview of the document, yet the overview may also distract the user. A study by Nekrasovski et al.

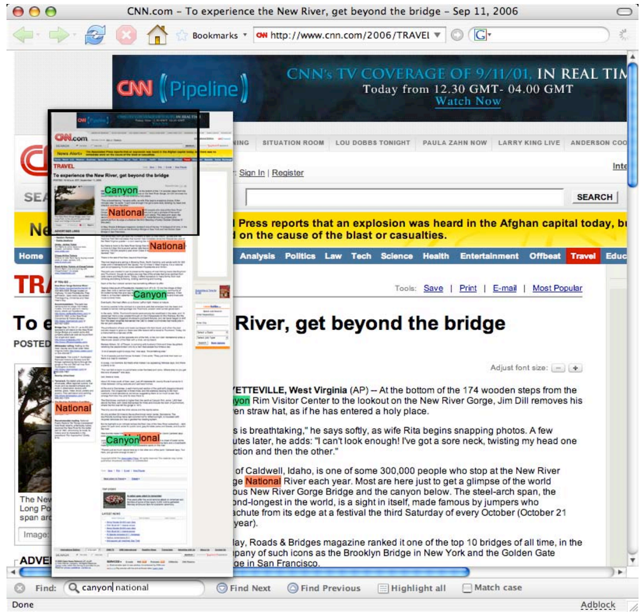


Figure 1: Sketch of transient overview of document with popout instances of words entered in the 'Find' bar.

[13] showed no performance effects of overviews used for navigating large hierarchical trees, but participants perceived them as beneficial. These results indicate that a transient overview may support particular uses such as searching or providing structural overview for navigation with less risk of distracting the user compared with a permanent overview.

3.2 Planning and Navigating Routes in Maps

Viewing and interacting with maps has received much attention in HCI research, and have been addressed by different visualization approaches including panning and zooming, overviews, and distortion techniques such as fisheye views [7]. A common use of maps is for planning and navigating a route to a destination. When navigating toward a remote destination, travelers commonly use a detailed map to orient themselves at their current location. However, an overview of the route to the destination may occasionally be needed to support a sense of direction and awareness in travels ahead. Getting an overview using conventional map applications may require considerable zooming in and out and panning the map to find road names and landmarks on the route ahead.

The mock-up presented in Figure 2 shows a way to extend a conventional map view with a transient visualization to address this problem. The user invokes the visualization by clicking on the route, calling up a map of a higher scale, thus showing the route farther toward the destination. In Figure 2, the user has further clicked three times on the route, to call up maps of continually higher scales, until the complete route is revealed. Finally, the visualization can be dismissed by clicking on the original map. The route provides fixing points for "stitching together" the maps of different scales, and the selected route can also be used to deduce contextual information, such as road names along the route that should be highlighted.

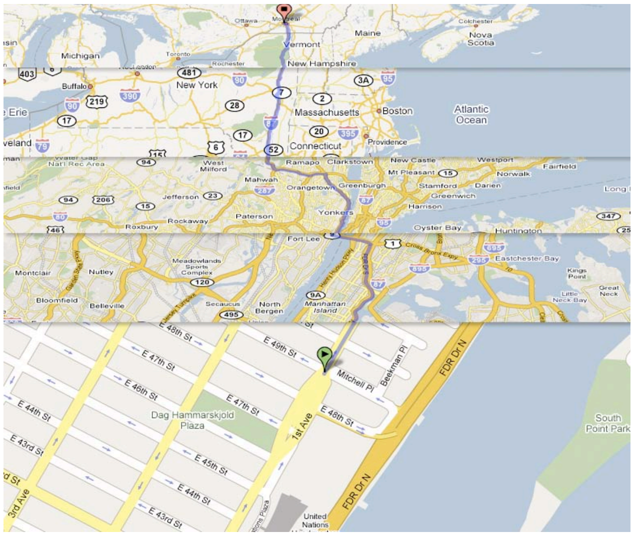


Figure 2: Route visualization where transient zoom-outs at progressively higher scales of a map have been called up by clicking repeatedly on the route to show the way to the destination.

This example shows how to extend the design space of information visualizations to transient use in a particular context, where permanent visualization techniques do not seem useful. Fisheye interfaces that geometrically distort maps are useful only to a limited degree of magnification and the distortion may inhibit users from recognizing shapes of roads and locations of landmarks [18]. Overview+detail interfaces may give an overview of the route, but to discern landmarks and road names along the route the user has to move the detail view. Zooming interfaces require the user to pan or continually zoom out to get an overview of the route, and then zoom in to see details of the route. In summary, while these different techniques may be useful for frequent contexts of use, it may be worthwhile to pursue transient visualizations for particular tasks such as the focus of this example.

3.3 Programming

Programming is a complex human activity that information visualizations potentially can support [11]. However, as mentioned earlier, applying a permanent visualization to source code can be complicated. Figure 3 shows a transient fisheye view of source code implemented as a plug-in for the Eclipse Java IDE. The visualization is invoked using a keyboard shortcut; popup views then appear above and below the editor window. The views contain lines with references to the variable that the user has currently selected with the caret. Arrow keys are used to select a line and pressing ‘Enter’ centers the view on the selected line. The visualization can be dismissed with the ‘Escape’ key.

Our design aims to support source code navigation and program understanding by providing lightweight access to contextual information relevant to the current focus in the source code. Compared with a permanent fisheye view, our design allows a large view of source code that programmers seem to prefer for writing and editing code. Furthermore, we aim to support fluid interaction with the transient fisheye view in programming by extending existing uses of the keyboard. A recent user study of programmers has shown extensive use of keyboard shortcuts for navigating in source code [13], and transient views showing outlines and type hierarchies are familiar in common programming environments such as Eclipse. We thus believe programmers may easily adopt transient visualizations that are invoked using keyboard shortcuts.

3.4 Writing

A very common task in writing is to find the right word at some point in a sentence. A thesaurus can be particularly effective for this task when writing in a language different from your mother tongue. In many word processors, finding the right word involves selecting a word, looking it up in the dictionary or thesaurus, browsing the definition and navigating links to synonyms.

Figure 4 shows a mock-up of a transient thesaurus visualization overlapping a text that the user is editing. The visualization is called up with a keyboard command to show words that are related to the word at the caret position. The user can interact with the visualization to explore more synonyms of a particular meaning; the highlight box can be moved with the cursor keys or mouse to one of the connected words, and selecting a word animates the visualization to center around that word, thus revealing more synonyms of that word. Also, the user can call up a window with the definition of a selected word. Finally, the visualization can be dismissed either to replace the original word in the sentence with the selected word (e.g., by hitting Enter) or without making changes to the text (by hitting Escape).

Our design utilizes the hierarchical organization of words in a thesaurus. In contrast to a linear textual representation, users can visually perceive from the visualization how synonyms of a word are grouped by similar meanings. Also, synonyms are presented close to the word and its surrounding text so that users can imagine how other words fit into the text. Finally, by making the visualization easy to invoke and dismiss, we aim for the use of the transient thesaurus to become an effortless part of writing.

4. RELATED WORK

We have aimed to demonstrate alternative uses of information visualizations by extracting and refining ideas from previous

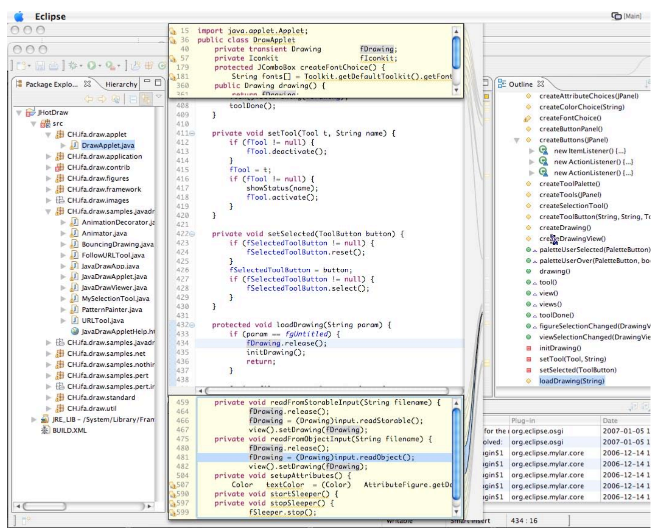


Figure 3: Prototype of transient fisheye view of source code that shows context information in popup windows above and below the editor window.

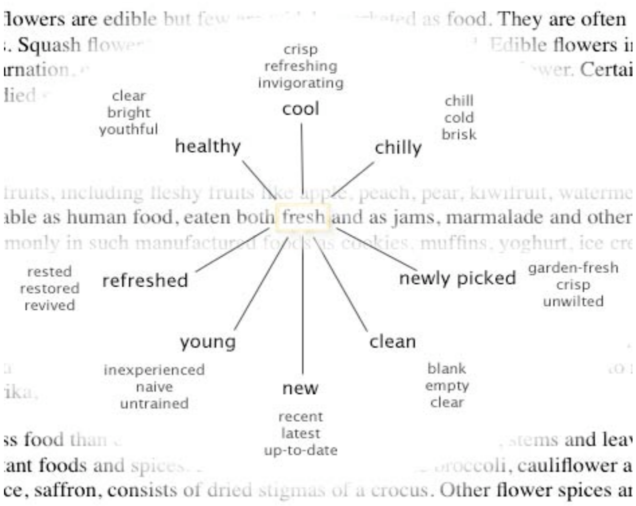


Figure 4: Transient thesaurus called up to show synonyms for the word "fresh" in a word processing application.

work. This section overviews such related work in HCI research that use transient representations of information and light-weight interaction.

Excentric labeling provides labels for a neighborhood of objects located around the cursor [5]. By showing labels temporarily when the cursor stays over an area for more than a second, the technique avoids information clutter and the need for extensive navigation. Side Views uses transient views to provide dynamic previews of multiple commands by visualizing the outcome of commands on the current selection, for instance using bold, italic or underline on selected text [16]. Zellweger et al. [19] studied the impact of lightweight, animated glosses for link anchors on hypertext browsing. Altogether, these transient preview techniques help users to probe relevant information without navigation and display switching, and to assess possible actions without resorting to "trial-and-undo".

Context menus that pop up near the mouse cursor or text caret present commands related to the current focus (e.g., for changing the font of selected text). Hotbox extends context menus with multiple menu bars centered around the cursor and with access to additional menus via mouse gestures [12]. See-through tools are another technique that provides close and contextual access to commands without requiring permanent display space [4].

Many information visualizations use brushing to highlight (or affect) instances in other views of an object that the user brushes over [2]. Highlighting techniques have been adopted, for example, in the Eclipse Java source code editor, where the caret can be placed in a variable to highlight all references in the code to that variable. Similar ideas have been demonstrated in spreadsheets [9]. These techniques provide immediate and non-intrusive visualizations through lightweight interaction.

Large and small displays accentuate problems in human-computer interaction, which have prompted HCI research to generate novel interaction techniques to temporarily bring objects that are otherwise hard to interact with closer to the user. The interaction technique called Vacuum helps reach remote objects through proxies that are transiently placed close to the cursor for easy manipulation, reducing the physical demands of the user [3]. Similar challenges in small displays have brought about techniques to visualize and navigate to off-screen targets with halos and proxies [10].

5. EXPERIMENT

To provide initial data about the usability of transient visualizations, we conducted a study comparing two interfaces for viewing maps, shown in Figure 5. Both interfaces include a view that can be panned to show different parts of a map; the user clicks and drags the mouse opposite the panning direction (i.e., the map follows the mouse). The interfaces also contain a semitransparent overview of the entire map. The overview partly covers the detail view so that it is possible but hard to discern map details in the detail view under the overview. However, it is not possible to "click through" the overview to interact with map details. Interaction with the overview differs between the interfaces. In the *Permanent interface (PI)*, the overview is permanently shown in the upper right corner of the detail view. The user can click and hold the left mouse button to drag a field-of-view box in the overview in order to pan the detail view. In contrast, the *Transient interface (TI)* does not permanently show an overview, but a transient overview can be invoked at the location of the mouse cursor by pressing and holding down the right mouse button; the overview appears so that the mouse cursor's location in the field-of-view box corresponds to the cursor's location in the detail view. Moving the mouse pans the detail view, and the overview disappears when the mouse button is released. Our primary goal is to compare the Permanent interface and the Transient interface. Therefore, we do not aim for our study to be realistic, but try to tease out differences in how users interact with the two interfaces.

5.1 Participants

20 students (4 female) at the authors' department participated in the experiment. The participants were between 21 and 50 years old (M = 29.3, SD = 7.9).

5.2 Tasks

Two types of task were used in the experiment. Both tasks involve maps of randomly placed circles with random names and randomly connecting lines. Maps are generated to resemble social networks. Colored circles are randomly scattered in the map, requiring participants to move the detail view to see them.

The first task type involves *selection* of 10 red circles in the map by finding and clicking on them. The selection task is designed to make participants alternate between navigating and interacting with objects in the map. Our hypothesis is that participants are slower with the Permanent interface, because they must move the mouse cursor between the overview for navigation and the detail view for clicking on circles, whereas in the Transient interface, the overview can be invoked and used immediately without first moving the mouse cursor.

The second task type involves *comparison* of 5 blue circles in the map and clicking on the smallest of them. The comparison task makes participants navigate and compare the size of blue circles at different locations in the map. We do not expect the Transient interface to have an advantage over the Permanent interface in this type of task. First, participants do not alternately navigate and interact with objects in the map; participants can navigate continually to the blue circles to compare them. Therefore the closeness of the transient overview is not important. Second, the overview may cover blue circles in the detail view that participants must see to compare their size. Although the overview covers part of the detail view in both interfaces, the fixed corner position of the permanent overview may help participants learn to consistently move blue circles into the visible part of the detail view. In contrast, invoking the transient overview at different positions can make it harder for participants to consistently move blue circles into view. However, participants can simply dismiss the transient overview to get a clear view of a blue circle when it has been located.

Since the overview used in this experiment shows the entire map, large maps result in a higher zoom factor than small maps. We varied the size of the maps used in the tasks to investigate the effect of varying zoom factors and varying distances between colored circles used in tasks. First, selection tasks with large maps require more precision in mouse movement when interacting with the overview. For example, the field-of-view box is smaller at higher zoom factors, which makes it is harder to move the mouse cursor from the detail view and target precisely in the permanent overview. Thus, we expect participants to perform worse with the Permanent interface compared with the Transient interface in tasks with large maps. Second, multiple red circles may be visible simultaneously in the detail view if the map is small, whereas large maps require participants to move the detail view to show each of the red circles in turn. As a result, the cost of targeting the mouse pointer in the permanent overview increases. Consequently, we expect participants to complete tasks faster with the transient overview in selection tasks with large maps compared with small maps.

5.3 Materials

Participants used a MacBook Pro laptop computer with an optical wireless mouse for the experiment. The screen was set to a 1440 x 960 resolution, and the size of the window containing the map interface was 700 x 700 pixels. Participants were guided through the experiment by a task view to the left of the interface window.

Two sizes of maps were used in the experimental tasks: small maps of 2000 x 2000 pixels (containing 200 circles) and large maps of 5000 x 5000 pixels (containing 600 circles). In small maps, two or three red circles may be visible simultaneously in the detail view, whereas only one red circle may be visible in large maps.

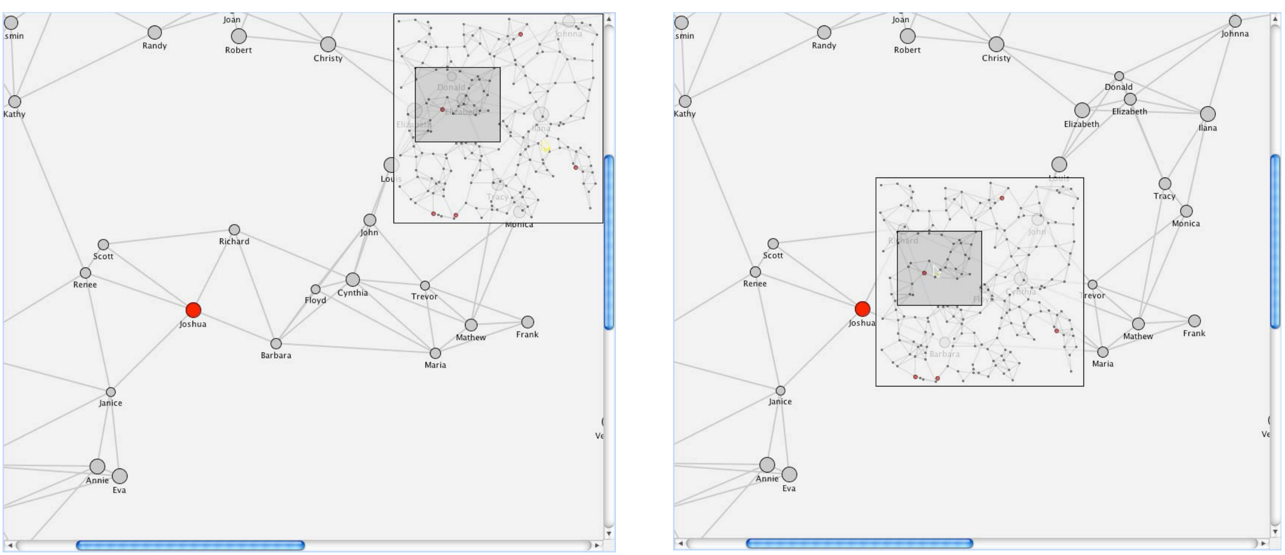
5.4 Design

We used a repeated measures design where four factors are varied within-subjects: interface type (PI, TI), size of the overview (O_{small}, O_{large}), task type (selection, comparison), and map size (small, large). Participants performed a set of 16 tasks with each interface. The order of interface and overview size was systematically varied across participants. The order of task type and map size for the eight tasks performed with each interface and overview size was also systematically varied. Thus 32 tasks with randomly generated maps were used; eight tasks for each combination of task type and map size.

We used two sizes of overviews because the size of the overview may affect the usability of the two interfaces. We expect participants to prefer a small overview in the Permanent interface because it covers a smaller part of the detail view compared to large overview. In contrast, a large transient overview does not permanently cover part of the detail view, so we expect participants to prefer a larger overview to a small overview in the Transient interface. The small overview used is 25% the width of the detail view and the large overview is 40% of the width of the detail view.

5.5 Procedure

Initially, participants were given an introduction lasting about ten minutes. In the introduction, participants were explained how to use the two interfaces and given a few minutes to try them. Next in the introduction, participants performed 16 warm-up tasks; four selection-tasks with PI, four selection-tasks with TI, four comparison-tasks with PI, and four comparison-tasks with TI.



Permanent interface (PI) Transient interface (TI)

Figure 5. The interfaces used in the experiment contain (left) a permanent overview in the upper-right corner and (right) a transient overview that is only visible when the right mouse button is pressed.

Participants performed two sets of tasks, one with each of the two interfaces. The participants were told to complete tasks correctly as quickly as possible. Following each set of tasks, a questionnaire about the interface just used was administered to the participants. The questionnaire contained six questions from QUIS [5] and five questions specific to the concerns of the experiment. A third questionnaire was administered after all tasks had been completed, asking the participants for their age and gender. The questionnaire also included three questions asking participants to compare the Transient interface with the Permanent interface on a comparative scale: first participants were asked which interface they preferred in general, then participants were asked which interface they found most appropriate for each type of task. Finally, participants were asked to write benefits and drawbacks of each interface and other comments. The entire experiment lasted between 30 and 45 minutes for each participant.

6. RESULTS

The results of the experiment consist of task completion times, accuracy and participant satisfaction. Of the 640 tasks that were completed across conditions, 13 tasks were discarded. First, due to an error in the experimental setup, two participants performed duplicate tasks and we discarded eight repeated tasks (two with TI, six with PI) because of possible learning effects. Second, we discarded three tasks (all with PI) where participants mistook a compare task for a selection task and clicked on the first blue circle that was visible. Third, two outlier tasks (both with TI), which either took more than 60 seconds for selection tasks or 30 seconds for compare tasks, were discarded.

6.1 Task Completion Times

Average completion times for the tasks are summarized in Table 1. We expected that participants would complete selection tasks faster using the Transient interface compared with the Permanent interface. In contrast, we did not expect comparison tasks to be performed faster with the Permanent interface. However, there was no significant difference in task completion times with the two interfaces for either type of task, $F(1, 19) = .293$, ns.

6.2 Accuracy

All of the selection tasks were completed correctly. In contrast, 273 of 310 comparison tasks were answered correctly. Accuracy is summarized in Table 2. Participants answer more tasks correctly with a large overview than a small overview, $F(1, 19) = 6.32$, $p < .05$. However, we find no influence of interface type on accuracy, $F(1, 19) = .812$, ns.

6.3 Satisfaction

Overall, participants preferred the Transient interface compared with the Permanent interface ($t = 3.387$, $df = 19$, $p < .005$), with 16 participants preferring the Transient interface and only four participants preferring the Permanent interface. There was no significant difference in what interface participants perceived to be most appropriate for selection tasks ($t = 2.070$, $df = 19$, $p > .05$) or comparison tasks ($t = 1.761$, $df = 19$, $p > .05$), although participants tended to prefer the Transient interface for both task types.

Table 2. Accuracy in comparison tasks for different interfaces, overview sizes and map sizes.

	Permanent interface			Transient interface		
	O_{small}	O_{large}	Avg.	O_{small}	O_{large}	Avg.
Small map	82.5%	97.4%	89.7%	90.0%	91.9%	90.9%
Large map	84.6%	97.2%	90.7%	80.0%	86.8%	83.3%
Average	83.5%	97.3%	90.2%	85.0%	89.3%	87.1%

Average satisfaction scores for the two interfaces are summarized in Figure 6 for the eleven questions that the participants answered. Overall, participants scored the Transient interface higher as assessed by multivariate analysis of variance, Wilk's lambda = .421, $F(1, 19) = 3.00$, $p < .05$. The results confirm our expectations that a transient overview reduces mental and physical efforts required of the user compared with a permanent overview. We had hypothesized that participants would prefer a large overview in the Transient interface and a small overview in the Permanent interface, but there was no significant difference between the interfaces in the size of overview that participants preferred.

6.4 Interaction Patterns

We analyzed the interaction data logged during the experiment to uncover differences in the use of the two interfaces. In selection tasks, interaction alternated between using the overview to bring circles into view and clicking on circles in the detail view. We expected the Transient interface to help participants complete these tasks with less mouse movement compared with the Permanent interface. To investigate this, we summed the distances that the mouse pointer traveled between mouse button events. Distance was calculated as the diagonal between screen coordinates of the mouse pointer. There was a substantial difference in the average distance per task for the two interfaces; a decrease of 60% from the Permanent interface to the Transient interface. Thus, the Transient interface appears to have reduced the sensory-motor efforts of the participants.

In comparison tasks, participants navigated between blue circles in the map to compare their sizes. The overview covered part of the detail view, especially in the large overview condition. Thus participants had to move the detail view, or dismiss the overview in the Transient interface, to get a clear view of the circles.

Table 1. Task completion times in seconds for different interfaces, overview sizes and task types.

		Permanent interface			Transient interface		
		O_{small}	O_{large}	Average	O_{small}	O_{large}	Average
Selection tasks	M	30.2	28.6	29.4	29.5	29.2	29.3
	SD	6.0	5.0	5.6	6.4	6.8	6.6
Comparison tasks	M	13.3	12.9	13.1	12.6	12.7	12.7
	SD	5.3	3.5	4.5	4.2	4.5	4.3
Average	M	21.8	21.0	21.4	21.0	21.2	21.1
	SD	10.2	9.0	9.6	10.0	10.0	10.0

Interestingly, participants mostly completed the tasks using the transient overview by continuously holding down the mouse button while navigating between the blue circles to compare them (in only 20 of 160 tasks, participants invoked the transient overview more than once). However, informal observations showed that participants using the Transient interface sometimes had trouble moving the blue circles clear of the overview—they did not dismiss the transient overview to get a clear view of the circle.

In the Permanent interface, most participants mainly clicked in the overview to move the detail view, a mode of interaction not supported in the Transient interface. Only three out of 20 participants dragged the field of view box as the main way of moving the detail view, which was the interaction mode also supported by the Transient interface.

7. DISCUSSION

The study reported in this paper provides initial insight into the general benefits of transient visualizations. We used tasks that focus on navigation to tease out differences between the interaction with the transient and with the permanent overview. In all, the results of our study suggest that having immediate and close access to the overview reduces sensory-motor efforts of the user. Surprisingly, we did not find this to reduce task completion times and error rates.

Even though participants preferred the Transient interface and completed the tasks with less mouse movement by accessing the overview immediately at the location of the cursor, they did not complete selection tasks faster. It seems that whereas the Transient interface helps moving red circles into the detail view, it does not help in acquiring the circles with the mouse. It is hard to move the map precisely using the overview in either interface: participants must make fine adjustments to position a target close to the overview if not move the mouse farther to acquire the target. However, compared with the Permanent interface where the overview is placed in a corner of the detail view, it is possible that positioning part of the map into the detail view demands more effort when the transient overview appears at different screen locations.

Some limitations must be considered when interpreting the results. Maps used were limited to sizes that allowed the entire map to fit in the overview. Larger maps require overviews with multiple levels of magnification. Furthermore, we focused on simple navigation tasks and participants used the interfaces for only a short period of time. Thus, our findings may not reflect varied, long-term use of the overviews. Additionally, three problems detracted from the usability of the Transient interface. First, we saw participants struggle with the overview when invoking the overview near the border of the detail view, making the overview only partly visible. Four participants commented on this problem in the questionnaire. Second, an implementation problem caused the transient overview to “stick” to the mouse cursor when dragging the field-of-view box out of the window, requiring participants to click in the detail view to make the overview disappear. Third, the data describing the interaction with the Permanent interface suggests that participants preferred to click in the overview to navigate in the map. However, the Transient interface only allowed users to drag the field-of-view box, because the overview was only visible while holding down the mouse button. Support for both interaction modes might improve the usability of the Transient interface. Toggling the

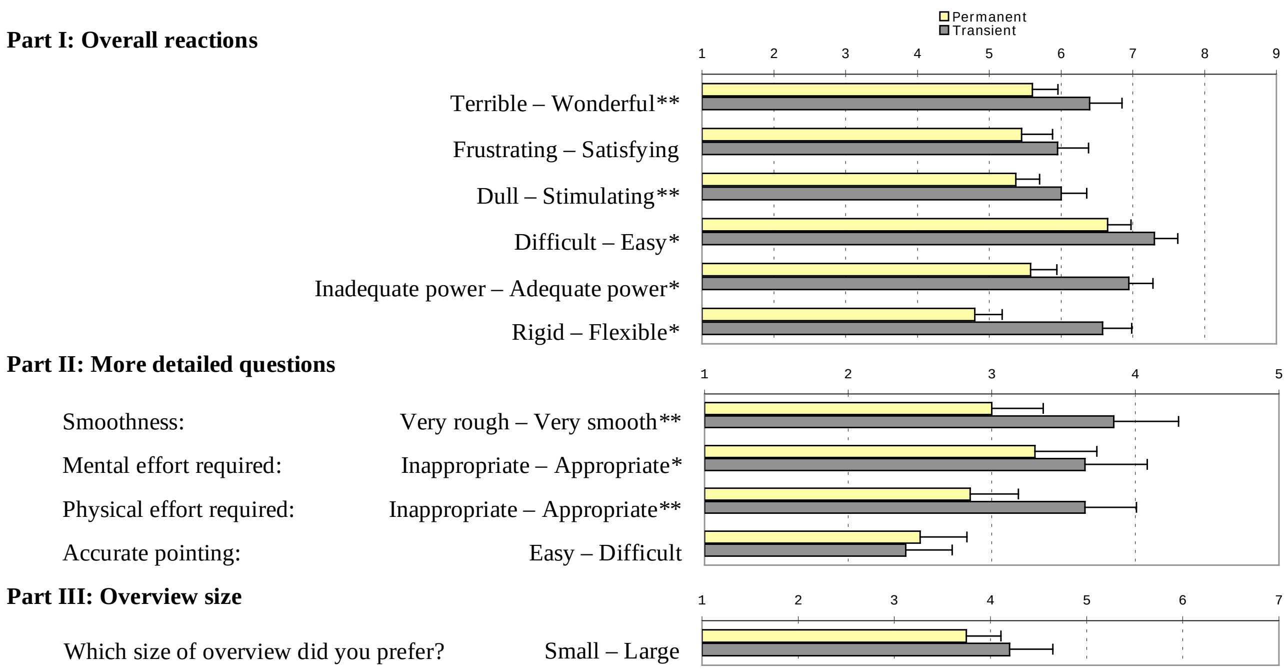


Figure 6: Average satisfaction scores (and standard error of the mean) for the eleven questions for the two interfaces. The anchor points on a semantic differential scale are shown for each question. Asterisks denote questions where the Transient interface scored significantly better (* = *p* < .05, ** = *p* < .01).

transient overview when the right mouse button is pushed is one possible solution.

More work is needed to further understand the general benefits and limitations of transient visualizations. Specifically, in the examples of transient visualizations presented in this paper, we have suggested the usefulness of transiently presenting contextual information related to the user's focus. Empirical evidence is needed to support this claim.

In complex work activity, transient visualizations may be useful to support sporadic tasks for which permanently changing the visual structure of information in the interface can impede frequent tasks. Studies are needed to understand what types of task that transient visualizations are suitable for. Evaluation of our transient fisheye view of source code may provide insights into the use of transient visualizations in expert tools.

Finally, conditions that limit the use of transient visualizations need to be examined. For example, transient visualizations that give no hint about their use are not accessible to novice users. Also, design and evaluation of transient visualizations must take into account that users may need longer practice time to make effective use of them compared to permanent visualizations that more readily afford their use.

8. CONCLUSION

We have characterized transient visualizations as interaction techniques that make immediate and transient use of information visualization close to, and in the context of, the user's focus of attention. In summary, transient visualizations offer a way of utilizing information visualizations to support specific contexts of use without making a permanent change to the user interface. We have presented examples of transient visualizations to support tasks in different domains.

To uncover how immediacy, transiency and closeness translate to actual and perceived improvements in the user experience, we conducted an experiment with map interfaces containing overviews. The results did not show significant improvements in time and accuracy with a transient overview compared to a permanent overview. However, our data suggest that tasks were performed with less sensory-motor efforts of the user, and 16 of the 20 participants preferred the transient overview.

Further studies are required to examine the general benefits and limitations of transient visualizations, to understand what types of task that transient visualizations are suitable for, and to provide design guidelines. Our initial data, however, suggest that transient visualizations may be useful, and that they are preferred by users to give immediate and close access to overviews.

REFERENCES

[1] P. Baudisch, B. Lee, and L. Hanna. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. *Proc. AVI '04*, 133–140, 2004. ACM Press.

[2] R. A. Becker and W. S. Cleveland. Brushing Scatterplots. *Technometrics*, volume 29, 127–142, 1987.

[3] A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. *Proc. CHI '05*, pages 361–370, 2005. ACM Press.

[4] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. *Proc. SIGGRAPH '93*, 73–80, 1993. ACM Press.

[5] J. P. Chin, A. Virginia, and K. L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proc. CHI '88*, 213–218, 1988. ACM Press.

[6] J.-D. Fekete and C. Plaisant. Excentric labeling: dynamic neighborhood labeling for data visualization. *Proc. CHI '99*, 512–519, 1999. ACM Press.

[7] K. Hornbæk, B. B. Bederson, and C. Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, 9(4):362–389, 2002.

[8] K. Hornbæk and E. Frøkjær. Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. *Proc. CHI '01*, 293–300, 2001. ACM Press.

[9] T. Igarashi, J. D. Mackinlay, B.-W. Chang, and P. T. Zellweger. Fluid Visualization of Spreadsheet Structures. *Proc. VL '98*, 118-125, 1998. IEEE Computer Society.

[10] P. Irani, C. Gutwin, and X. D. Yang. Improving selection of off-screen targets with hopping. *Proc. CHI '06*, 299–308, 2006. ACM Press.

[11] M. R. Jakobsen and K. Hornbæk. Evaluating a fisheye view of source code. *Proc. CHI '06*, 377–386, 2006. ACM Press.

[12] G. Kurtenbach, G. W. Fitzmaurice, R. N. Owen, and T. Baudel. The Hotbox: efficient access to a large number of menu-items. *Proc. CHI '99*, 231–237, 1999. ACM Press.

[13] G. C. Murphy, M. Kersten, and L. Findlater. How Are Java Software Developers Using the Eclipse IDE? *IEEE Software*, 23(4):76–83, 2006.

[14] D. Nekrasovski, A. Bodnar, J. McGrenere, F. Guimbretière, and T. Munzner. An evaluation of pan & zoom and rubber sheet navigation with and without an overview. *Proc. CHI '06*, 11–20, 2006. ACM Press.

[15] B. Suh, A. Woodruff, R. Rosenholtz, and A. Glass. Popout prism: adding perceptual principles to overview+detail document interfaces. *Proc. CHI '02*, 251–258, 2002. ACM Press.

[16] M. Terry and E. D. Mynatt. Side views: persistent, on-demand previews for open-ended tasks. *Proc. UIST '02*, 71–80, 2002. ACM Press.

[17] A. Woodruff, A. Faulring, R. Rosenholtz, J. Morrison, and P. Pirolli. Using thumbnails to search the web. *Proc. CHI '01*, 198–205, 2001. ACM Press.

[18] A. Zanella, M. S. T. Carpendale, and M. Rounding. On the effects of viewing cues in comprehending distortions. *Proc. NordiCHI '02*, 119–128, 2002. ACM Press.

[19] P. T. Zellweger, S. H. Regli, J. D. Mackinlay, and B.-W. Chang. The impact of fluid documents on reading and browsing: an observational study. *Proc. CHI '00*, 249–256, 2000. ACM Press.