

TrailMap: Facilitating Information Seeking in a Multi-Scale Digital Map via Implicit Bookmarking

Jian Zhao

Daniel Wigdor

Ravin Balakrishnan

Department of Computer Science

University of Toronto

{jianshao, dwigdor, ravin}@dgp.toronto.edu

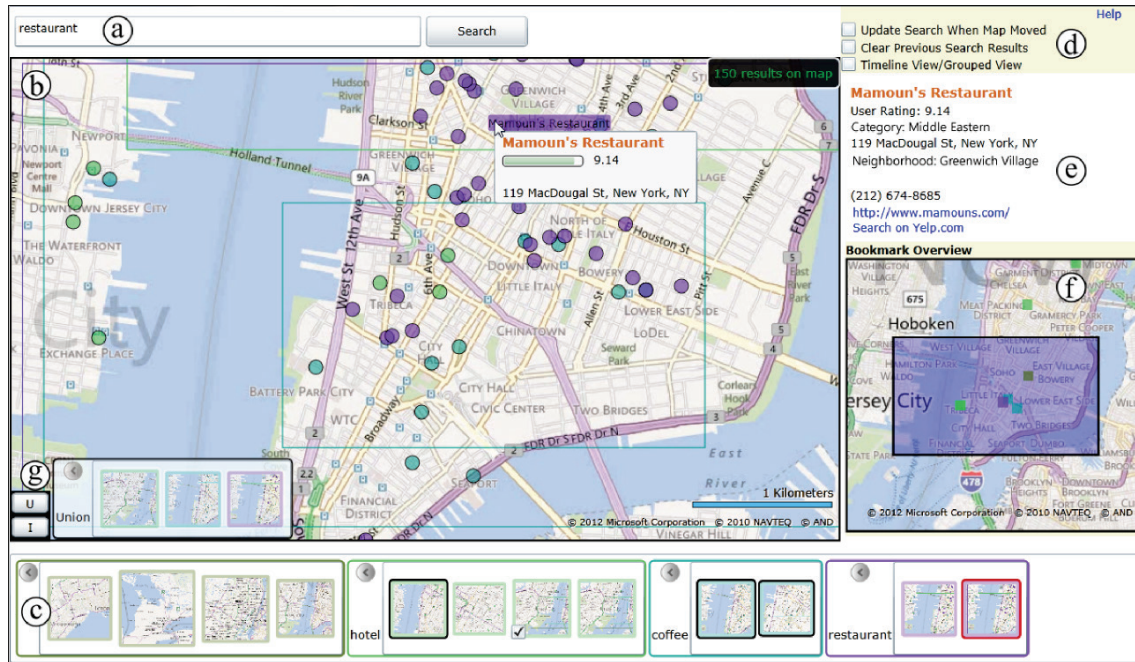


Figure 1. A user is doing trip planning with TrailMap by searching different businesses with (a) the search box. While the user is exploring this area in (b) the main map view, automatically generated bookmarks are displayed in (c) the bookmark panel. With (g) the advanced bookmark revisiting panel, he is showing the search results of three different bookmarks and is trying to find an optimum solution. He can also view more information with (e) the business detail panel and (f) the map overview.

ABSTRACT

Web applications designed for map exploration in local neighborhoods have become increasingly popular and important in everyday life. During the information-seeking process, users often revisit previously viewed locations, repeat earlier searches, or need to memorize or manually mark areas of interest. To facilitate rapid returns to earlier views during map exploration, we propose a novel algorithm to automatically generate map bookmarks based on a user's interaction. TrailMap, a web application based on this algorithm, is developed, providing a fluid and effective neighborhood exploration experience. A one-week study is conducted to evaluate TrailMap in users' everyday web browsing activities. Results showed that TrailMap's

implicit bookmarking mechanism is efficient for map exploration and the interactive and visual nature of the tool is intuitive to users.

Author Keywords

Digital map browsing; revisit; implicit bookmarking; multi-scale exploration; interaction history.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces – Interaction styles.

INTRODUCTION

Online map services, such as Google Maps and Bing Maps, offer an efficient and convenient way for users to interact with geographical information. Websites designed for exploring local areas, which include mapping capabilities, have become increasingly popular, such as Yelp (www.yelp.com) for local business reviews and recommendations, RE/MAX (www.remax.com) for real estate listings, and Priceline (www.priceline.com) for hotel bookings and travel planning.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

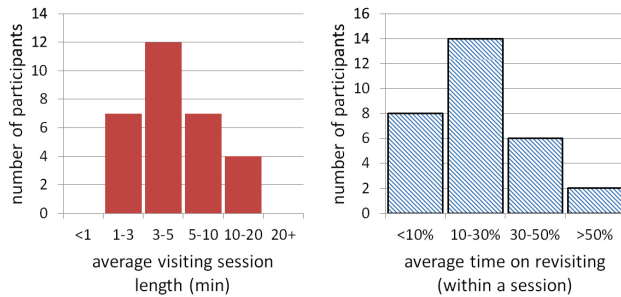


Figure 2. Preliminary survey results.

As users perform information seeking tasks in maps (such as choosing a restaurant), it is common for them to backtrack by revisiting locations and repeating earlier searches. Despite the convenience of online map exploration, revisitation can be time consuming and frustrating. We conducted a preliminary survey with 30 participants from local universities regarding their map browsing habits (Figure 2). Nearly half of the participants reported spending between 10% and 30% of their map browsing time finding earlier *sites* to revisit (a term we will use to encompass map position, zoom level, and search results). An additional 27% of the participants reported spending more than 30% of their browsing time on these activities. Similar phenomena of revisitations are observed in other tasks. For example, Tauscher *et al.* [28] found that 58% of individual web visits were actually revisits.

To reduce the effort inherent in revisitations, map users may create *explicit* bookmarks (i.e., manually make a bookmark) to assist their search and bring them quickly back to areas of interest. Typically, users will open new browser tabs for selected businesses or add them to a customized map, such as My Places in Google Maps. However, explicit bookmarking imposes an additional cognitive burden on users, interrupts the interaction flow, and requires the user to anticipate the revisitation [2].

The notion of *implicit* bookmarking, which *automatically* generates potential bookmarks for revisitation while the user is viewing the information, has been shown to be effective in application domains such as document reading [4, 16] and web browsing [29]. Nevertheless, little work has been done to explore automatic bookmarking in information seeking during geospatial exploration.

The challenges to this process are threefold: 1) the information space is continuous, with no explicit boundary of defining a *site*, making it difficult to recognize likely revisitation positions; 2) the navigation happens in a multi-scale 2D space with heterogeneous interactions between sites of interests (i.e., 2D pan and zoom); and 3) the map’s meta-information content is dynamic and varies with the user’s keyword searches.

To address these challenges, we propose a novel algorithm for implicit bookmarking within information-seeking tasks during multi-scale exploration on digital maps. Additionally, we introduce TrailMap, a web application prototype which

aligns the proposed algorithm with Bing Map APIs. Using TrailMap, bookmarks are automatically generated based on a user’s interactions, such as mouse clicks and wheel movements, eliminating the need for manual bookmarking and repeat searches. Finally, we contribute the results of a one-week evaluation of TrailMap with users in their real map browsing environments and discuss improvements of the algorithm and the interface.

RELATED WORK

Revisitation and Interaction History

Usage patterns of current computer-based interfaces are highly repetitive, such as command use [13], menu and list selection [8, 14], document reading [4], web browsing [3, 9], and visual analytics [15]. To make interactions more effective, researchers have proposed various techniques to support revisitations, ranging from user-performed manual operations, to automated interface updates, based on a recorded interaction history.

Explicit bookmarking technologies require a series of intended user actions to mark interesting information. For example, in Adobe Reader, adding a bookmark to the bookmarks panel; whereas in Google Maps, clicking the “save to map” link when the user wants to add a pushpin to My Places. In document reading, Bookmark Scrollbar [22] places user-created bookmarks on a standard scrollbar. However, explicit bookmarks have 3 major limitations: 1) manual marking operations require extra cognitive effort by the user; 2) interaction flows are interrupted; and 3) users are required to predict which sites they will want to subsequently revisit [2].

These issues have led to the development of implicit bookmarking techniques, such as the “Forward/Back” buttons on web browsers and Microsoft’s Visual Studio. Nevertheless, although these upgrades have improved the efficiency of revisitations, they still fall short of a truly intuitive interaction. The visiting history list is not visible on these interfaces, and users often misunderstand the purpose of the “Back” operation and become frustrated when a desired item cannot be found [9]. Visiting histories are typically hidden behind other controls, such as requiring the user to navigate to the “Recent Documents” menu in Word and the “History” tab of a web browser. Footprints [29] and PageLinker [27] offer more intuitive ways of displaying the visiting history, through the use of maps, paths, and annotations. However, this technique is specific to exploration tasks in a discrete information space.

Some implicit bookmarking systems have been proposed for a continuous information space, such as document reading. The Read Wear system shows reading histories of an entire document through the use of histograms to display the number of edits and the frequency of readings on a scrollbar [16]. Footprints Scrollbar maintains a recency list of visiting locations within a document and displays them on the scrollbar [4], and Visit Wear improves revisitation in

distorted fisheye visualizations by highlighting previously visited items in both discrete and continuous datasets [26]. As well, Chun *et al.* developed an implicit bookmarking algorithm for document reading tasks by segmenting reading blocks from users’ mouse-wheel events [32]. However, all of these systems are constrained to bookmarking in a linear presentation space or with simple forms of interactions, such as 1D scrolling.

Moreover, the information content of tasks like document reading is generally static, whereas in some tasks—such as visual analytics with information visualization systems—a user’s exploratory visit dynamically modifies the visual representation. A revisitation tool which retrieves previous visualization states can facilitate data analysis and user communication [15]. The introduction of tools such as GraphTrail [10] and KronoMiner [33], have enhanced implicit bookmarking by presenting and organizing a graphical history of visualization in a hierarchical structure. However, here too, the information space (i.e., states of visualizations) is limited to discrete exploration.

Geospatial exploration combines the most difficult elements of these challenges: the information space is continuous, multi-scale, the content is dynamic, and the navigation methods are heterogeneous. To our knowledge, the prior art has failed to solve the problem of implicit bookmarking for information seeking on digital maps.

Multi-Scale Exploration

Furnas *et al.*’s space-scale diagram portrays the conceptual framework of multi-scale exploration [12], in which the scale is on the z -axis and the 2D space is laid-out on the x , y -axes. The image space on the u , v -axes is scaled along the z -axis (presuming that the viewport remains the same size during the exploration).

The use of navigation techniques with multi-scale interfaces has been extensively studied. Zooming is a commonly used technique which usually employs the use of a specific widget or the mouse wheel. Some systems initiate zooming based on the selection of an object [30] or a region [24] on the screen. Semantic zooming, which dynamically modifies the selection and structure of data to be displayed, is widely used in current online maps [6]. To increase zooming efficiency with a very large information space, some techniques have been proposed to automatically scale the content based on a user’s interactions, such as the speed of panning [18], the distance to the scrollbar [5] or the degrees of interest of the information content [19].

A major limitation of zooming interfaces is known as the “desert fog” problem because of the absence of orientation features at high-level scales [21]. Studies indicate that adding a view of the entire world (i.e., the overview+detail techniques) can benefit a user’s navigation and overcome the “desert fog” problem [17]. Commercial systems, such as the outline panel of Microsoft PowerPoint and the view panel at the bottom-right corner of Google Maps, have

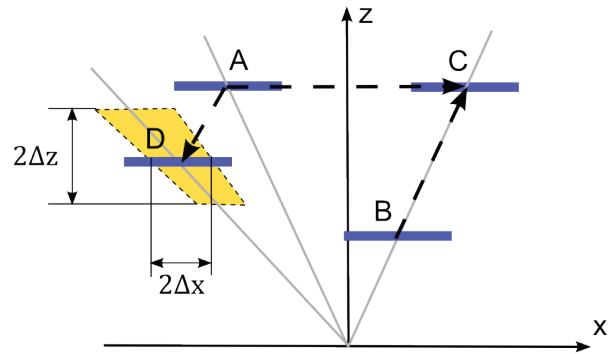


Figure 3. Pan and zoom in space-scale diagram.

already adopted this technology. Javed *et al.* have proposed a more advanced technique, called PolyZoom, for exploring a large 2D space by creating multiple levels of views on different scales of the information space [20].

A third multi-scale exploration method introduces the focus+context technique, using distortion to seamlessly combine the region of focus, shown in great detail, with the surrounding context. The Mac OS dock panel, the well-known fisheye views [7] and rubber sheet [25] fall in this category. However, it can be difficult to acquire targets with a distorted focus at a very high scale. Pietriga *et al.* explored the usage of time and translucence to achieve more efficient interactions in focus+context interfaces [23].

However, none of these multi-scale exploration techniques have yet incorporated the aspect of interaction history, which we believe is very important, in information seeking with maps. Although PolyZoom can, to some extent, track the exploratory path of browsing a multi-scale 2D map, it requires a *de facto* explicit bookmarking technique which demands a user’s attention and effort [20]. An exception is the Glimpse technique, which considers all navigation actions temporary until confirmed by an explicit action. Once confirmed, however, the technique does not include a method to revisit previous views [11].

TASK MODEL AND ALGORITHM

The strength of a novel algorithm for implicit bookmarking lies in its ability to recognize useful sites that a user would like to revisit. Here, we introduce a conceptual model for information-seeking interactions with maps, and propose an algorithm for implicit bookmarking based on that model.

User Interaction Model

As Figure 3 indicates, the space-scale diagram [12] conceptualizes “pan” and “zoom” interactions, which are the main navigation operations in multi-scale exploration. From a geometrical point of view: 1) pan is represented by horizontally shifting the viewport (e.g., A→C), 2) zoom is mapped to the viewport movement along the ray of connecting the origin and the center of the viewport (e.g., B→C), and 3) other movements indicate a combination of pan and zoom (e.g., A→D).

The task of information-seeking on a multi-scale map can be viewed as a continuous interaction stream, consisting of

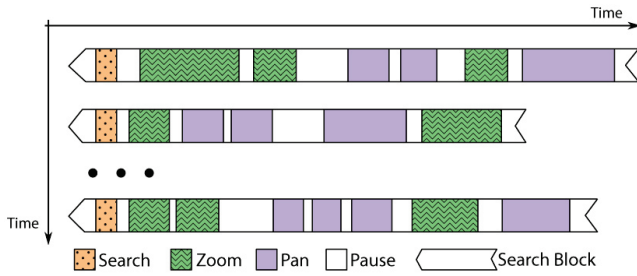


Figure 4. User interaction model.

search, zoom, pan and pause operations, each lasting for a certain time period. Since the information content is dynamic, (i.e., the items displayed on the map change each time the user initiates a new keyword search), we can intuitively divide the task into several individual interaction blocks with relatively stable information content. Each search block commences with a search, followed by a series of pan, zoom and pause actions of varying time lengths (Figure 4). The characteristics of these interaction blocks, such as the durations, distances and types of the navigation operations and pauses, are used to find intuitive bookmarks in the interaction stream.

Implicit Bookmark Generation

Our goal is to determine the most logical and useful sites to generate implicit bookmarks that the user is likely to revisit. A bookmark in this continuous and dynamic data space should include the keyword search, the viewport location and zoom level (where the search takes place). Therefore a revisit is defined as a visit to a previous viewport with an identical search (i.e., redo the query).

Based on the above user interaction model (Figure 4), we define three heuristic opportunities for implicit bookmark generation. The first is to create a bookmark when a user initiates a new search (thus displaying newly queried results on the map).

Second, bookmarks should occur when a user visits different places, providing the waypoints during the multi-scale exploration. Long distance zooming and panning, which can be measured using the geometrical concepts in Figure 3, are possible delimiters of consecutive visits to bookmark. It is worth noting that pan and zoom are interactions with distinct natures that are often imposed by different input devices; panning one pixel on the screen does not appear to present the same level of task difficulty for the user as zooming one level of the scale. In this project, we treat pan and zoom separately, i.e., setting two different thresholds, D_p and D_z , respectively for identifying long pans and zooms.

Third, bookmarks should be generated during specific pauses—a longer dwell time may indicate that the user has a higher interest in browsing a particular location. For example, Footprints Scrollbar [4] uses a two-second dwell time threshold for document reading tasks with 1D scrolling. In our work, we consider 4 different pause states based on a

user’s interactions prior to, and following, a dwell time: pan-pan, zoom-zoom, pan-zoom, and zoom-pan. It is unclear whether these dwell times have similar characteristics because of the difference between zoom and pan. The “transition points” (i.e., of pan-to-zoom and zoom-to-pan) could also be potential interesting positions (and distinctly different, as segmentation of actions of the same type may be caused by clutching the input device). Thus, we introduce 4 thresholds: T_{PP} , T_{ZZ} , T_{PZ} , and T_{ZP} , which use dwell times to identify ideal opportunities to bookmark.

Algorithm

We propose the following algorithm for implicit bookmarking in information seeking with multi-scale maps.

1. On startup, set the *current* keyword as *null* (for collecting bookmarks that are with no queries).
2. When a *search* operation is detected, generate a bookmark and set the query as the *current* keyword.
3. When a *zoom* operation is detected, generate a bookmark with the *current* keyword if:
 - a. The zoom distance is larger than D_z .
 - b. Or, the dwell time before it was longer than T_{ZZ} or T_{PZ} , based on the previous operation (zoom or pan).
4. When a *pan* operation is detected, generate a bookmark with the *current* keyword if:
 - a. The pan distance is larger than D_p .
 - b. Or, the dwell time before it was longer than T_{ZP} or T_{PP} , based on the previous operation (zoom or pan).
5. When a new bookmark is generated, delete existing bookmarks in a close geographic space-scale proximity that contain the same keyword, (i.e., bookmarks within $[-\Delta z, \Delta z]$ in scale and $[-\Delta x, \Delta x]$ in spatial distance), as illustrated by the yellow area in Figure 3. This operation is designed to avoid clutter and prevent the creation of too many bookmarks in a small geographical area.
6. When a bookmark is selected, navigate the map view to the recorded viewport and redo the query with the recorded keyword and set it as the *current* keyword.

Threshold values depend on the input device, as well as users’ input habits. In our TrailMap implementation, the parameters were selected empirically based on an in-lab pilot study with 4 volunteers (see Implementation Parameters section). During the pilot study (about 30-45 minutes), participants tried TrailMap with different parameter settings on their own computers. Participants provided feedback about the performance, allowing the experimenter to adjust and finalize the parameter values.

THE TRAILMAP INTERFACE

Using the algorithm described above, we developed TrailMap, a research prototype that uses the Bing Map services [1] under the Silverlight platform. TrailMap consists of 7 interactive interface components (Figure 1): (a) a search box, (b) a main map view, (c) a bookmark panel, (d) a configuration panel, (e) a business detail panel, (f) a map overview, and (g) a logical bookmark revisiting panel. Next, we describe the visualizations and interactions encountered in the TrailMap interface.

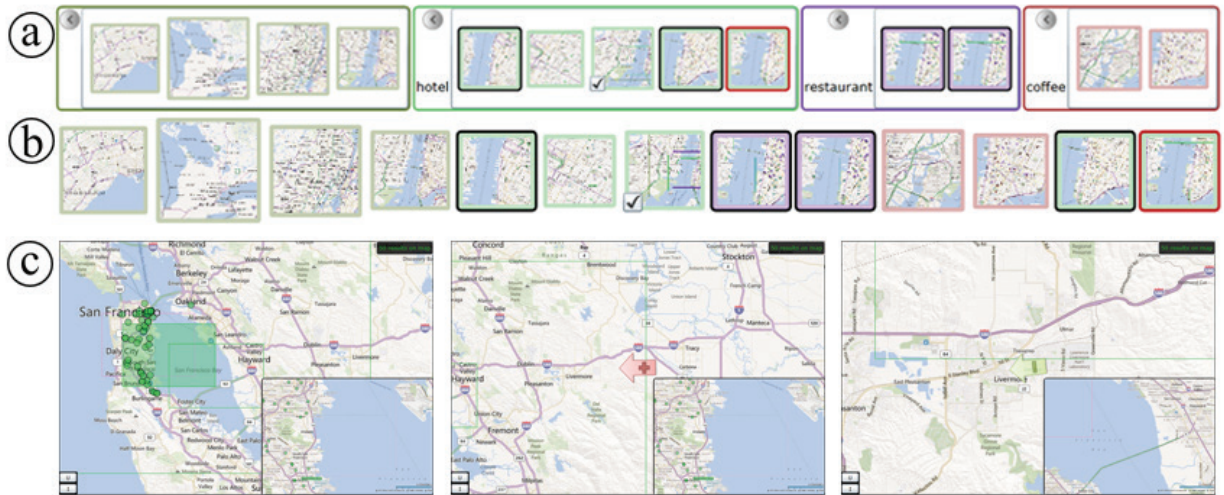


Figure 5. Bookmark visualizations: (a) grouped view, (b) timeline view, and (c) bookmark indicators (left: the bookmark can be seen in the viewport; middle and right: zooming and panning operations needed to navigate back the bookmark).

Bookmark Visualizations

Visualizing bookmarks is an interesting challenge. Footprints Scrollbar utilized the 1-dimensional space of the scrollbar [4]. Given the 4-dimensional space of our bookmarks, however, the implicitly-generated bookmarks of a particular map viewport with a specific search keyword are presented in several ways on different UI components.

Bookmark Panel

Indications of sites. In the bookmark panel (Figure 1c), a bookmark contains a thumbnail image of the recorded map viewport. The image size is related to the scale factor, where a smaller image size correlates to a higher zoom level. Bookmarks associated with the *current map viewport* are outlined in red. Bookmarks generated from *previous map viewports* are outlined in two ways: if the viewport location is still visible from the current viewport, the outline color is black; if the viewport location is not visible from the current viewport, there is no outline at all.

Grouped view and global timeline. By default, bookmarks are color-coded and grouped by associated keywords in the bottom bookmark panel, where the first always contains bookmarks with no associated search terms, i.e., null keyword. Within a group, bookmarks are sorted by the time they were created, beginning with the least recent from left to right (Figure 6a). Alternatively, a user can choose to display TrailMap bookmarks along a global timeline. This feature allows bookmarks with different search terms to be co-mingled, allowing users to track the search exploration history across time (Figure 5b).

Main Map View

As Figure 5c-left indicates, when a user hovers over a bookmark item, a shaded rectangle representing the bookmarked viewport is displayed in the main map view (Figure 1b). Simultaneously, a larger snapshot of the bookmarked map image is overlaid in the bottom-right corner of the main view. For a bookmark whose map viewport is located outside of the current viewport, an

arrow indicating its relative direction and zoom level is displayed, in place of the shaded rectangle. The arrow is shown as a red plus for zooming-in or green minus for zooming-out (Figure 5c-middle and -right). These features allow the user to interactively explore the history without committing to a change of viewport.

Map Overview

The map overview (Figure 1f), allows a user to explore all the bookmarked map viewports, represented as color-coded rectangular pins. Hovering over a pin initiates the same effects on main view as described above. The semi-transparent blue rectangle indicates the current viewport shown in the main view (Figure 1b). The map overview is synchronized with the movements of the main map view when a user is interacting with the main view; but when a user pans or zooms the overview to see other map regions, the main view is not affected.

Bookmark Basics

An initial TrailMap bookmark is created when a user initiates a keyword search in the search box. TrailMap sends queries via Bing Map APIs and displays the resulting local businesses within the current map viewport as circular pins on the map (Figure 1b). When a user *hovers* over a circular pin, a tooltip and a name label for more details appear. If a user *clicks* a circular pin, complete information about the business is shown on the side panel (Figure 1e).

As a user explores the local area using pan and zoom, bookmarks with corresponding sites (viewport locations and search keywords) are automatically generated and displayed in the bottom bookmark panel (Figure 1c). If a user *clicks* a bookmark, TrailMap navigates the viewport back to the associated location and sends queries with the recorded keyword again.

For each keyword, the system retains up to a predefined number of bookmarks in a queue (see Implementation Parameters section). Bookmarks are time-stamped and the

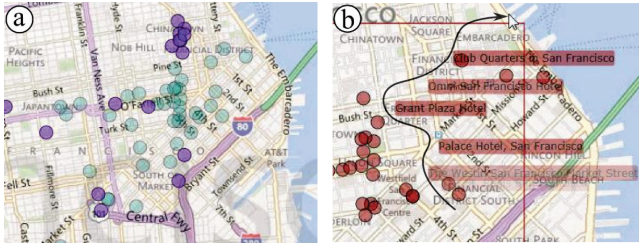


Figure 6. Semi-transparency effects for short-term memory: (a) last search results are translucent and (b) name label fade-out animations as the cursor is moving.

oldest one is deleted from the queue once the number of bookmarks goes over. A user can *fix* a bookmark by ticking a checkbox (see the 3rd item in “hotel” group in Figure 1c) to override the delete command. In addition, a user can manually delete a bookmark, or a batch of associated bookmarks that use the same keyword.

Advanced Bookmark Revisitations

TrailMap allow users to easily perform advanced searches by quickly merging the information from multiple bookmarks, by using *union* and *intersection* functions.

To activate these features, a user clicks one of the “U” (union) or “I” (intersection) buttons located in the bottom-left corner of the main map view (Figure 1g). A user selects the desired bookmarks, one by one, to generate the advanced view on the map. Each bookmark brings its own associated query results, i.e., a keyword search within a particular map viewport, to take part in the union or intersection operation with current map view.

In *union* mode, search results associated with each selected bookmark will be added to the map. To illustrate, in Figure 1 the user has chosen three bookmarks: the 3rd one in “hotel”, the 1st one in “restaurant”, and the 1st one in “coffee”. TrailMap produces a combined map which displays all the businesses matched with the three queries together in the viewport encompassing the search areas. This feature may be useful for users to execute trip planning where multiple categories of businesses are important.

In *intersection* mode, two or more bookmarks are combined to produce a map that displays only the common elements of the queried results of the bookmarks. The intersection feature is ideal for using multiple bookmarks to narrow down a search for particular businesses with multiple desired attributes. For example, the intersection of a bookmark for “pizza” and a bookmark for “bar” will produce a map displaying results for businesses that belong to both bookmarks: in other words, a bar that serves pizza.

Additional Exploration Features

In addition to bookmark generation, TrailMap offers 2 other mechanisms designed to assist the map exploration. First, each time a new search is initiated, TrailMap displays local businesses seen in previous search results with semi-transparent circular pins (Figure 6a). These pins help a user

to recognize and remember the previous search. Second, when a user hovers over a circular pin, a name label pops up. Once the cursor leaves the pin, the business name label fades, but remains visible over the course of a 2-second fade-out animation to prompt revisits. For example, in Figure 6b, the user moves the cursor along the black trail. These visualizations help the user recall the previous search results and the items that he has recently visited.

Implementation Parameters

Based on the pilot study, we chose the following parameter values for the TrailMap bookmarking algorithm.

Thresholds for long-distance pan and zoom are: $D_z=2$ scale-levels and $D_p=0.75 \times \text{mapview's diagonal length (pixels)}$; the parameters for dwell times are: $T_{zz}=1$ second, $T_{pz}=T_{zp}=1.6$ seconds, and $T_{pp}=3$ seconds; and the parameters for determining nearby bookmarks are: $\Delta z=1.5$ scale-levels and $\Delta x=0.6 \times D_p$ pixels.

An important issue for consideration throughout this work was the computation of pan and zoom distances. The start and end points of a pan can be easily tracked via mouse-down and mouse-up events, respectively. However, start and end points for a zoom are more difficult to pinpoint, as mouse-wheel events are uniform (i.e., no start-zoom or end-zoom events provided). In TrailMap, we set a threshold for delimiting continuous mouse-wheel events, such that a new zoom operation is detected when the time between two successive mouse-wheel events exceeds 60 milliseconds.

Based on work by Alexander *et al.*, which shows that a recency list of 8 items covers approximately 80% of revisitations in document reading [4], TrailMap permits a maximum number of 8 bookmarks per keyword group. Further analysis on the ideal number of bookmarks for map navigation scenarios is left for future work.

EVALUATION

The goal of our study is to evaluate the usefulness of TrailMap for supporting map information seeking and exploration tasks with implicit bookmarks during users’ every day map browsing interactions.

Participants

We recruited 11 participants (4 females), aged 20-31, from our local university community through mail lists. The participants came from different academic programs including arts, engineering, economics, and science. Based on their self-reports from pre-study questionnaires, the participants normally accessed existing online map applications 6-10 times per week for an average for 5-10 minutes each time.

Apparatus

Participants accessed TrailMap through an IIS server from their own desktops/laptops. Participants used IE, Firefox, Chrome, or Safari running on Windows/Mac machines. Screen display sizes ranged from 13 to 24 inches diagonal.

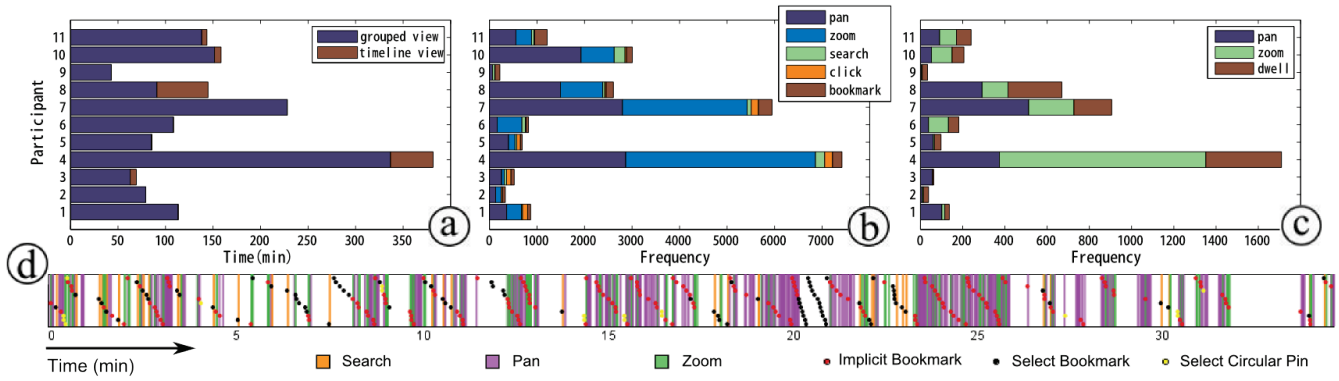


Figure 7. Visualizations of interaction logs: (a) overall TrailMap usages, (b) user operations by type, (c) auto-generations of implicit bookmarks by user interaction, and (d) interaction log data of Participant 4’s session 3.

Procedure and Design

Participants used TrailMap for 1 week on their own computers. Throughout the study period, interaction data and events, (i.e., mouse clicks, mouse-wheel events and viewport movements) were captured and sent to the server. Participants were instructed to use TrailMap for their normal everyday activities, as much as possible, instead of using similar existing applications. Prior to the beginning of the study, participants received a 30-minutes individual tutorial session in our lab on how to use TrailMap and how to access it from different platforms. A detailed user instruction manual and its associated web link were also provided. At the end of the study, participants completed a written questionnaire and participated in a face-to-face interview to provide feedback about their experiences using TrailMap. Each participant received \$30 for participation.

RESULTS

Interaction Logs

We collected a total of 95 visits of TrailMap, with an associated 1,556.4 minutes of usage and a mean session length of 16.4 minutes. Figure 7a shows an overview of the total usages of every participant (mean 141.5 minutes).

We observed a total of 837 keyword search operations, 33,324 mouse events for map manipulations on the main view, and 1,322 interactions with the bookmark panel. Figure 7b provides a further decomposition of user operations, indicating participants performed a fairly amount of interaction with bookmarks compared to clicks and searches. Of all the bookmark operations, 1,025 were simple bookmark revisits, 229 were advanced bookmark queries (111 for union, 118 for intersection), and 81 were bookmark grouped/timeline view changes.

Within all captured mouse events, there were 22,218 mouse down/up events, 11,106 mouse-wheel events, and 718 mouse clicks on the circular pins displayed on the map. Based on total mouse events, we identified 11,033 pans, of which 1,592 were long pans, and 9,661 zooms, of which 1,638 were long zooms, according to our algorithm implementation. Altogether TrailMap generated 4,439 implicit bookmarks (including auto-deleted ones) from

Question	Mean	SD
Q1. Overall I was satisfied with the design of this application.	7.9	0.9
Q2. The interface of visiting and manipulating bookmarks was easy to use and intuitive.	8.1	1.4
Q3. The interface of exploring search results (e.g. fading-out labels) on the map was helpful.	8.1	1.2
Q4. The automatically generated bookmarks were helpful in map exploration and search tasks.	8.3	1.3
Q5. The functions of logically revisiting multiple bookmarks were helpful.	8.3	1.3
Q6. I'd like to have a plugin of auto-bookmarking functions with the existing online map applications.	8.9	1.3

Table 1. Questionnaire results.

these long zooms and long pans together with 1,209 long dwells detected by the algorithm (Figure 7c). The interaction logs of a sample session are shown in Figure 7d, which further ensures our proposed interaction model and indicates a user’s frequent bookmark revisiting activities.

Questionnaire Results

In the post-study interview, participants completed a questionnaire of 6 questions, using a 1-10 Likert scale (from strongly-disagree to strongly-agree). As Table 1 indicates, TrailMap received an overall rating of 7.9 for participants’ satisfaction with the application’s design (Q1); the 2 main TrailMap’s new features (i.e., implicit bookmarks and advanced bookmark revisitations) as well as TrailMap’s interface design (i.e., interactions and visualizations) all received relatively high mean scores—above 8 out of 10 (Q4 and Q5, Q2 and Q3). Participants ranked their desire to have TrailMap’s features available to use with existing online map applications as an 8.9 out of 10, which is very encouraging

User Feedback

In this section, we report the qualitative feedback gathered from post-study interviews and examine these comments in light of data provided from the TrailMap interaction logs. Participants are referred as “P<participant number>”.

Use of the Basics of Implicit Bookmarks

TrailMap users executed a bookmark revisitation every 70 seconds on average during the trial period (see Figure 7d as an example session). Participants indicated that they relied

on the bookmarks during their information-seeking tasks with online maps and they found the interaction to be intuitive and enjoyable.

“I was finding good driving schools that are close to my home or a subway station or the campus. After looking at them [on the map] for a while, I was getting lost and I couldn’t find where my home or my school is. Using the bookmarks, I could easily see where I was from and even go back and forth. So I don’t need to zoom out all the way and then zoom in again to see my positions.” (P1)

We observed that a new keyword search was conducted every 1.8 minutes and 63% of the searches were revisited via bookmarks. Participants commented that the keyword searches, captured in the automatic bookmarks, were helpful in neighborhood exploration,

“Last Saturday, my wife and I wanted to go out for dinner. We searched Indian [food] then Thai then Indian again and then to something else. So I really liked that you could have all the stuff together and just clicks let you go back [to previous locations and searches]. Once I made a decision I just pinned the bookmark and kept searching.” (P6)

P11 found it *“really fun”* to explore coffee shops near the campus and her home, as she mentioned that *“it was super easy to switch between locations without memorizing them”*.

Use of the Union and Intersection of Implicit Bookmarks

Interaction logs indicated that 17% of all bookmark operations were the advanced bookmark functions (i.e., showing the union or intersection of collected bookmarks). On average, advanced functions were used every 6.6 minutes. Participants stated that TrailMap’s ability to combine multiple sets of search results was practical for travel and errand planning.

P3 used TrailMap to assist her travel next week by searching hotels, restaurants, car rentals and so on. *“First, you don’t need to remember them [while you are searching] and you can easily re-do the search. Second, the union is very helpful because you can see the distributions [of all the search results]. So I can choose the best combination.”* (P3)

“I used the union [operation] to see if there were McDonalds, supermarkets and drugmarts nearby [a bank], then I could figure out which branch is good for me. It is much much easier than the Google Maps where I have to click ‘search nearby’ every time for every keyword.” (P4)

Some participants commented that TrailMap’s intersection feature was very convenient in customizing a map search to find businesses that met specific needs. For example, participants reported using the intersection feature to locate: *bars that have steak* (P10), *a carwash with vacuums* (P3), and *sushi places that also have ramen* (P11).

“My friend wanted to go for spicy Thai curries but I just wanted dim sum or other mild Chinese food. Then we searched Thai, Chinese [...] and tried to use the

intersection to look for some Asian fusion restaurants that are good for both of us. And it worked! It saved me a lot of time on clicking into each restaurant.” (P5)

Implicit Bookmarking Algorithm

On average, TrailMap generated a new implicit bookmark approximately every 20 seconds during the study period (see Figure 7d as an example). Roughly one-third (32%) of the bookmarks were utilized by participants to perform either simple revisits or advanced operations. In general, participants were satisfied with the implicitly generated bookmarks and thought that the creation caused by pan, zoom and dwell interactions were suitable for their different navigation habits.

P4 said *“they capture the purposes of your exploration”* and participants found that the pace and amplitude of their navigation operations could affect the bookmark generation, which was *“intelligent and reasonable”*. P3 mentioned that TrailMap successfully detected the neighborhoods she was interested in during the exploration of a new city.

It is known that every automatic algorithm cannot always be perfect. False positives, bookmarks that *are* created at a location that a user does not want to revisit, and true negatives, bookmarks that *are not* created at a spot where a user actually wants one, did occur. However, only P6 mentioned that *“sometimes too many bookmarks were generated”*. P9 and P2 felt that the implicit bookmarks were not always precise and suggested that integrating TrailMap with the traditional explicit bookmarking capabilities would be better for more general situations.

TrailMap Visualizations and Interactions

Participants were satisfied with the visual design of TrailMap, stating that the interface was intuitive, aesthetically pleasing and easy to interact.

Within the two bookmark items visualizations, the grouped view is primarily applied (92.3% of the total time, also see Figure 7a). P4 stated the grouped view is preferred, because it is *“more organized and easier to match the bookmarks with the items on the map by colors”*. However, P3 and P4 indicated that the timeline view is also a useful view for tracking the viewport movements over time.

Most participants appreciated the semi-transparency effects applied to previous search results and the fade-out animations of name labels, as they found that it assisted their exploration of search results on the map. P4 said *“[the fade-out effect] was lovely [...] You can have a sense of what you were interested in.”* P11 especially liked the fact that TrailMap encoded a bookmark item size with the zoom levels: *“it helps you find which bookmark you want to click, especially when you are exploring a larger area”*.

For future improvements, P8 suggested that placing geo-location tags (e.g., North America, Canada, or Manhattan) onto bookmark items in Figure 1c would be more helpful for navigation. Some participants mentioned that the

interactions within the timeline view could be improved by incorporating shortcuts, such as the left/right cursor keys.

TrailMap and Existing Practices

All participants agreed that TrailMap was useful in many aspects of their everyday map exploration and search activities. Several participants mentioned that the automatic bookmarking functions and the ability to logically combine search results on the map were “*hardly seen elsewhere*”, and sometimes they were “*surprised by the new map-browsing experience*” offered by these functions. Participants reported that, as they gained familiarity and confidence using TrailMap, they began to rely upon it and some enthusiastic users such as P4 and P7 “*searched almost everything with it*”.

One item of note is that P8 did not often use TrailMap for map-based searches; rather, P8 used it as a navigation tool for exploring geographical information of the map dataset. He mentioned “*[The bookmarks] not only showed me the navigation trails and where I lingered for a long time but also helped me easily switch among different places of interests. But that is very hard with Google Maps, with the My Place feature, as I usually do.*”

Apart from the implicit nature of TrailMap bookmarks, participants also appreciated the fact that these bookmarks recorded the viewport area and the search, when asked for comparing with other map applications they used. P10 said “*The ‘localized search’ is very convenient for exploring neighborhoods especially when you are new to the city, because you may be interested in different things at different geo-locations or scale levels*”. Some participants commented that the abilities of the bookmarks “*restoring both the view and the search results*” provided a more fluid and natural user experience.

Participants also identified several functions provided by the existing applications that were not offered in TrailMap, which resulted in an overall rating score that was lower than individual aspects of the system (Table 1). Participants commented that after finding a desired location using TrailMap, they would like to see the distance between 2 places and directions on how to get there. Also, participants felt the dataset of user ratings and reviews of businesses was somewhat incomplete (due to the Bing Map API). Though an external link of Yelp search was included in TrailMap, participants desired a more integrated interface.

DISCUSSION

This research aimed to develop a novel algorithm and an interactive web application to fill an existing gap in online map exploration.

Often, users find switching to a novel mechanism interrupts their established habits and patterns of interaction. Interaction logs revealed that the average usage of TrailMap was 141.5 minutes per participant in 1 week, exceeding the participants’ typical usage of online maps prior to the study. When viewed in light of the participants’ comments, this

finding suggests that TrailMap was interesting, intuitive and useful. Qualitative user feedback for TrailMap was positive. Participants actively used the implicit bookmarks to revisit previous locations and to perform advanced union and intersection functions to customize searches. TrailMap’s implicit bookmarks, together with different visualizations and the advanced bookmark revisitation functions provide an effective and fluid user experience of map explorations. P6 said, “*I never thought I would need the bookmarks because I’m too familiar with this city. But I realize I really do when I did a lot of searches and it simply works!*”

Interface Design

We note that TrailMap UI is more complicated than traditional online mapping tools, but we deal with complex map information-seeking tasks, where the data space is continuous, dynamic and multi-scale. Thus it is difficult to remove interface complexity while maintaining useful and advanced functionalities. Moreover, the study results showed that participants were able to use it with minimal instructions, indicating that the interface was intuitive enough and easy to operate by novice users.

Another potential issue of the TrailMap interface is the visual clutter, for example, P11 and P6 felt that frequent search and navigation interactions generated a lot of bookmarks and that a cluster of these rectangular outlines were difficult to work with on the main map view. They proposed to add a feature that allows users to hide the rectangles for particular bookmarks or bookmark groups. P11 added that it would be useful to have the option of auto-hiding the bookmark panel when the screen is small. Also, an integration of some target selection techniques such as Escape [31] can further improve the experience.

Algorithm Design

Both the frequency of revisiting bookmarks (once per 70 seconds) and the overall use of implicit bookmarks (recorded as 32%) suggest that frequent revisitations during map exploration are needed as well as the TrailMap’s algorithm is useful and satisfies participants’ different needs and browsing habits. Based on the study results, we further propose the following directions to improve the current algorithm which could potentially enhance the experience for a more generalized user.

First, TrailMap could be adapted to allow users to customize the granularity of bookmark generation. For example, a user could configure TrailMap to generate fewer bookmarks by adjusting the thresholds. Furthermore, it is straightforward to integrate the traditional explicit bookmarking along with TrailMap, in which the manual bookmarks could be generated just as a user fixing the bookmarks in the current implementation. Second, different users have different interaction behaviors, and different devices have different specifications on input events. Thus developing a more intelligent bookmarking algorithm—that can learn its parameters on-the-fly while the user is interacting with the map—could be suitable for general

situations, although our study showed that current implementation with static parameters was applicable. Third, our current algorithm associates bookmarks with keywords without any semantic considerations. Keywords direct the bookmark grouped view, user inputs and the deletion of nearby bookmarks. A potential improvement is to integrate natural language-processing algorithms into keyword classifications, such that hotel, lodging, and motel could all be classified as the same bookmark group.

CONCLUSION AND FUTURE WORK

We have presented an implicit bookmarking algorithm for information-seeking tasks with digital maps where the data space is continuous, dynamic and multi-scale. We developed TrailMap, a novel web application prototype by implementing the proposed algorithm with Bing Maps APIs. We conducted a one-week study in users' everyday map browsing activities to evaluate the effectiveness of TrailMap in users' everyday map exploration. We found that participants actively used TrailMap for their web-based map searches, and they appreciated the usefulness of the implicit bookmarking mechanism and the intuitive interaction and visualization of the tool which provided a fluid and effective map exploration practice.

There are several promising directions for future work with TrailMap, in addition to the interface and algorithm improvements discussed earlier. First, TrailMap could be adapted to add functions such as: exporting, saving, reloading, and even sharing, the bookmarks. Currently, TrailMap supports the user's short-term memory with implicitly-generated bookmarks and the visualizations of business-name labels. These additional functions would enhance a user's longer-term memory. Second, TrailMap could be expanded to offer a mobile version and to integrate multiple online data sources to improve the user experience. Third, TrailMap could benefit from long-term user studies to further evaluate the usability of this unique application and its mobile versions in the wild.

ACKNOWLEDGMENTS

We thank the participants in our study, Kim Wright for assistance with the paper, Stephanie Santosa for narration of the accompanying video, and all the DGP members for the valuable discussions.

REFERENCES

1. Bing map services. <http://msdn.microsoft.com/en-us/library/dd877180.aspx>.
2. D. Abrams, R. Baecker, and M. Chignell. Information archiving with bookmarks: personal web space construction and organization. In *CHI'98*, pages 41–48, 1998.
3. E. Adar, J. Teevan, and S. T. Dumais. Large scale analysis of web revisitation patterns. In *CHI'00*, pages 1197–1206, 2008.
4. J. Alexander, A. Cockburn, S. Fitchett, C. Gutwin, and S. Greenberg. Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar. In *CHI'09*, pages 1665–1674, 2009.
5. C. Appert and J.-D. Fekete. Orthozoom scroller: 1d multi-scale navigation. In *CHI'06*, pages 21–30, 2006.
6. B. B. Bederson and J. D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST'94*, pages 17–26, 1994.
7. M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *UIST'01*, pages 61–70, 2001.
8. A. Cockburn, C. Gutwin, and S. Greenberg. A predictive model of menu performance. In *CHI'07*, pages 627–636, 2007.
9. A. Cockburn and B. Mckenzie. What do web users do? an empirical analysis of web use. *Int. J. of Human-Computer Studies*, 54:903–922, 2000.
10. C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *CHI'12*, pages 1663–1672, 2012.
11. C. Forlines, C. Shen, and B. Buxton. Glimpse: a novel input model for multi-level devices. In *CHI '05 extended abstracts*, pages 1375–1378, 2005.
12. G. W. Furnas and B. B. Bederson. Space-scale diagrams: understanding multiscale interfaces. In *CHI'95*, pages 234–241, 1995.
13. S. Greenberg and I. H. Witten. Supporting command reuse: empirical foundations and principles. *Int. J. Man-Mach. Stud.*, 39(3):353–390, 1993.
14. C. Gutwin and A. Cockburn. Improving list revisitation with listmaps. In *AVI'06*, pages 396–403, 2006.
15. J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. on Vis. and Comput. Graphics*, 14(6):1189–1196, 2008.
16. W. C. Hill, J. D. Hollan, D. Wroblewski, and T. McCandless. Edit wear and read wear. In *CHI'92*, pages 3–9, 1992.
17. K. Hornb, 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.
18. T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *UIST'00*, pages 139–148, 2000.
19. W. Javed, S. Ghani, and N. Elmqvist. Gravnav: using a gravity model for multi-scale navigation. In *AVI'12*, pages 217–224, 2012.
20. W. Javed, S. Ghani, and N. Elmqvist. Polyzoom: multiscale and multifocus exploration in 2d visual spaces. In *CHI'12*, pages 287–296, 2012.
21. S. Jul and G. W. Furnas. Critical zones in desert fog: aids to multiscale navigation. In *UIST'98*, pages 97–106, 1998.
22. S. A. Laakso, K. P. Laakso, and A. J. Saura. Improved scroll bars. In *CHI '00: extended abstracts*, pages 97–98, 2000.
23. E. Pietriga and C. Appert. Sigma lenses: focus-context transitions combining space, time and translucence. In *CHI'08*, pages 1343–1352, 2008.
24. D. C. Robbins, E. Cutrell, R. Sarin, and E. Horvitz. Zonezoom: map navigation for smartphones with recursive view segmentation. In *AVI'04*, pages 231–234, 2004.
25. M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *UIST'93*, pages 81–91, 1993.
26. A. Skopik and C. Gutwin. Improving revisitation in fisheye views with visit wear. In *CHI'05*, pages 771–780, 2005.
27. A. Tabard, W. Mackay, N. Roussel, and C. Letondal. Pagelinker: integrating contextual bookmarks within a browser. In *CHI'07*, pages 337–346, 2007.
28. L. Tauscher and S. Greenberg. Revisitation patterns in world wide web navigation. In *CHI'97*, pages 399–406, 1997.
29. A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. In *CHI'99*, pages 270–277, 1999.
30. A. Woodruff, J. Landay, and M. Stonebraker. Goal-directed zoom. In *CHI'98*, pages 305–306, 1998.
31. K. Yatani, K. Partridge, M. Bern, and M. W. Newman. Escape: a target selection technique using visually-cued gestures. In *CHI'08*, pages 285–294, 2008.
32. C. Yu, R. Balakrishnan, K. Hinckley, T. Moscovich, and Y. Shi. Implicit bookmarking: Improving support for revisitation in within-document reading tasks. *Int. J. of Human-Computer Studies*, 71(3):303–320, 2013.
33. J. Zhao, F. Chevalier, and R. Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *CHI'11*, pages 1737–1746, 2011.