

# WebThumb: Interaction Techniques for Small-Screen Browsers

Jacob O. Wobbrock\*, Jodi Forlizzi\*†, Scott E. Hudson\*, Brad A. Myers\*

\*Human Computer Interaction Institute and †School of Design

Carnegie Mellon University, Pittsburgh, PA 15213 USA

{ jrock, forlizzi, scott.hudson, bam }@cs.cmu.edu

## ABSTRACT

The proliferation of wireless handheld devices is placing the World Wide Web in the palms of users, but this convenience comes at a high interactive cost. The Web that came of age on the desktop is ill-suited for use on the small displays of handhelds. Today, handheld browsing often feels like browsing on a PC with a shrunken desktop. Over-reliance on scrolling is a big problem in current handheld browsing. Users confined to viewing a small portion of each page often lack a sense of the overall context – they may feel lost in a large page and be forced to remember the locations of items as those items scroll out of view. In this paper, we present a synthesis of interaction techniques to address these problems. We implemented these techniques in a prototype, WebThumb, that can browse the live Web.

## Keywords

Handheld devices, small displays, interaction techniques, web browsing, thumbnails, zooming, kinetic typography.

## INTRODUCTION

The day is near when every handheld will wirelessly connect to the web and be perpetually online [6]. Though the earliest PDAs lacked processing gusto, the higher-end devices of today rival desktop machines of only six years ago and the power of today's desktops can be expected of handheld devices in the near future. However, this technology requires careful interaction designs to succeed. Desktop software will not be effective when merely crammed into a 320×240 handheld screen.

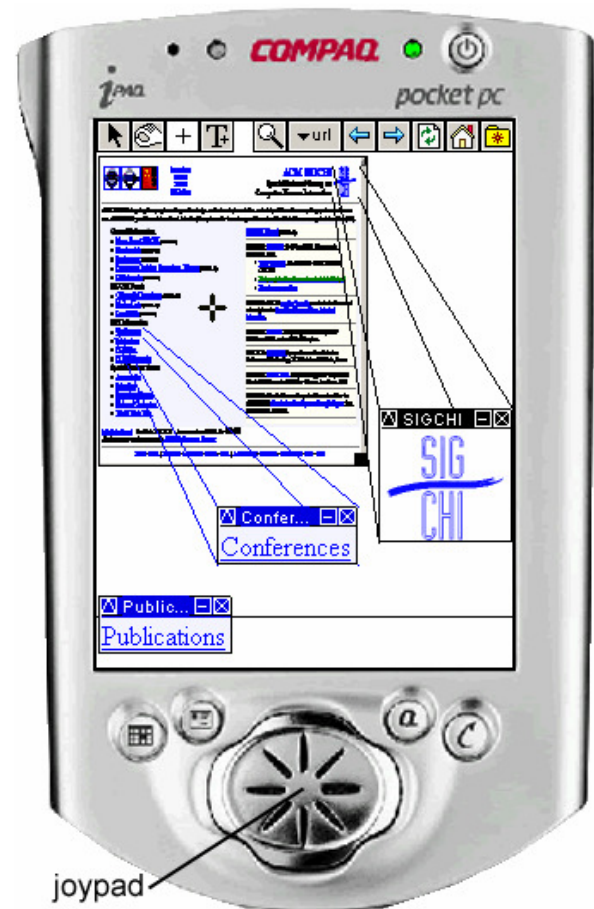
Various approaches for browsing on small screens have had problems. The Wireless Application Protocol (WAP) and Web Clipping Applications (WCAs) use proprietary languages to request portions of web pages they reformat for display. Users often end up with awkward content ill-suited for display on small screens.

Another problematic approach is dynamically transforming the HTML to be displayed on a handheld. These algorithmic approaches have had only mild success because pages “in the wild” are too complex and varied to be successfully transformed in the general case. Most pages

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.

UIST'02, October 27-30, 2002, Paris, FRANCE.

Copyright 2002 ACM 1-58113-488-6/02/0010...\$5.00.



**Figure 1.** (Actual size) WebThumb showing a thumbnail of <http://www.sigchi.org/> with three extracted elements, two of which have their “origin lines” on. The two lower items are links and can be retained and followed any time.

are designed with only a desktop audience in mind and may not lend themselves to repurposing as handheld content [5]. In the end, text can be shrunk and realigned, images scaled, and tables rebuilt, but pages lose their intangible qualities of layout, segmentation, and flow.

In this paper we offer a purely interactive solution. We leave the layout of pages as their authors intended (Figure 1), retaining the original design and avoiding the inevitable mangling of content endemic in algorithmic transformation. Unlike other approaches [1,11], our approach does not

touch the HTML but improves the browsing experience by enhancing the interaction techniques available to the user.

Specifically, our contributions are (a) allowing users to “pick up” and collect elements from a web page (Figures 1, 2), thereby extending interaction with a page beyond the viewing of that page; (b) solving the “lack of context” problem by using interactive thumbnails with zoom capabilities (Figures 1, 3); (c) employing a cursor as an interactive focal point (Figures 1, 3), something handhelds traditionally lack; (d) using *rapid serial visual presentation* (RSVP) of text [3,7] for reading unlimited text in a limited space (Figure 4); and (e) exploiting the joypad to pick up elements from a page in a cardinal direction from the cursor (Figure 5), a technique especially useful on dense pages. The technology that makes these techniques possible is the ability to programmatically relate to a page not solely as a stream of HTML, but also as a rendered grid of pixels and the underlying elements those pixels represent.

These contributions have been implemented in *WebThumb: Web Exploration using a Body of Techniques for Handheld Ubiquitous Mobile Browsing*. As shown in Figure 1, the WebThumb prototype is implemented in a desktop simulation of a Compaq iPaq (with accurate size and pixel count). This allows us to use mature implementation tools to rapidly explore and iterate techniques, to make use of an existing browser instead of implementing one from scratch, and to take advantage of the CPU speeds that we will see on a PDA in a few years but cannot get today.

We leveraged research from a variety of areas in designing WebThumb. Text-enhanced thumbnails of web pages were shown to be useful for conveying information in search engine results [11]. Thumbnailing was also used on the desktop to better manage documents, including web pages, in Data Mountain [8]. Our zoom feature is similar to that in SlideShow Commander [6]. Ideas from research into efficient use of small screen real estate proved helpful in our design [4]. Finally, our use of RSVP was inspired by results from research on display techniques for rapid reading [7]. Others have explored kinetic typographic techniques as a way of conveying meaning in desktop environments [2,3,9,10].

## INTERACTION TECHNIQUES IN WEBTHUMB

In this section, we describe the ways in which we attacked the problems associated with handheld browsing. Our solutions are embodied in the WebThumb prototype.


### Thumbnails and Tools

Thumbnails, sometimes called “graphical summaries”, are useful in conveying information quickly [11]. Users familiar with a web page on the desktop can often recognize it in thumbnail form. Microsoft Windows takes advantage of this by optionally displaying thumbnails of selected HTML files in the left margin of folder windows.

In WebThumb, users are able to see the overall context of a page in a resizable thumbnail (Figure 1). This contributes to a sense of orientation and does not presume that the upper-left corner of a page is the best place to start viewing, as

conventional handheld browsers do. Users are free to zoom into any part of the page without having to scroll to get there. When resized, the thumbnail shows the page as a desktop browser would, with wrapping and layout identical to that of a typical desktop browser.

Interaction with the thumbnail employs a set of four tools, available at the top-left of the interface. The tools are selected by tapping with the stylus or by pressing the two hard buttons to the left of the joypad, which cycle through the tools to the left or right.

With the arrow tool,  tapping on the thumbnail produces the same result that mouse-clicking does on the desktop. For example, tapping a link will navigate. In visiting relatively stable web pages familiar to them, users are able to tap links in thumbnail view without reading them, a positive use of spatial memory [8].

### Picking Up Elements from a Page

Usually some elements within a page are more interesting than others. However, users may not want to zoom into every item they find interesting from the thumbnail view. Furthermore, some elements, links in particular, can be useful beyond the page they appear on. For these reasons, we provide the ability to pick up items from the thumbnail and retain them in a larger, easily readable size. This can be

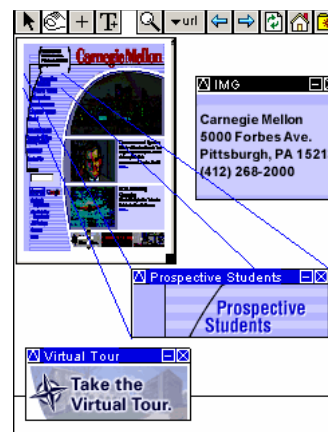



Figure 2. Picked up items from <http://www.cmu.edu/>.

Prospective students may wish to keep these items around while browsing other parts of the site. Title bars show ALT text for images, if available.

Title bars show ALT text for images, an indicative string for text and textual links, or, as a last resort, the tag name (e.g., “IMG”) of the extracted item.

Extracted elements must not consume much screen space. Therefore, these elements can be collapsed to their title bars with the “-” title bar button. Because the thumbnail is resizable only down to the horizontal black line at the bottom, collapsed elements can be saved below the line without being obscured (Figure 3).

thought of as an “inverse zoom” since elements *come out* rather than the user going in.


The “pick up” tool  extracts any element—a table cell or image, link or text field—into its own window (Figure 2). Extracted items that are links have blue title bars across the top; non-linked items have black title bars. Extracted elements that have links *in* them, e.g., tables of links, have black title bars but the links within them can be tapped and followed. Title bars

While an element's page is still visible, tapping on the upside down "V" at the left end of the element's title bar will display "origin lines" (Figures 1, 2, 5). These lines quickly identify a collapsed element and its place of origin on the thumbnail. Until toggled off, origin lines stay affixed to the thumbnail, even while elements are dragged.

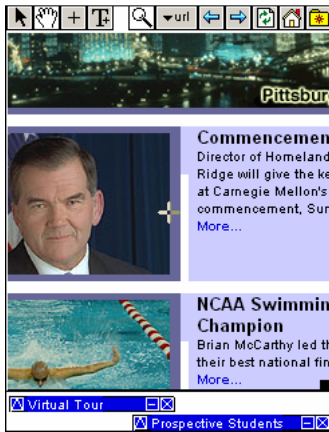
Picking up elements from web pages is particularly useful on handhelds because navigation is cumbersome and scrolling prevalent, making items more difficult to find [5].

### Targeted Zooming and Panning

Handhelds do not typically rely on a cursor to guide selection and keyboard input the way desktops do. However, we found it useful to employ a cursor-like device, the *target*, in order to provide multiple interactions.


The pick up tool, the crosshairs tool,  and the RSVP tool (described below) all place the target wherever they are used.

The user can "get a closer look" by zooming into the web page at the target point (Figure 3). Zoomable thumbnails mimic exploration in general: first gain a view of the whole, then focus on parts of highest interest.



**Figure 3.** When zoomed in, the thumbnail fills the view. Pan by moving the stylus across the page (while the hand tool is selected) or by using the joypad.


Without the target to correlate items in the two views, zooming feels less accurate.

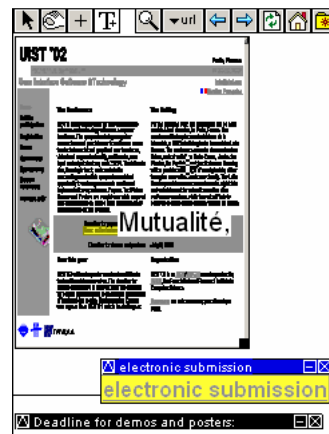
When zoomed in, the thumbnail automatically maximizes to fill the screen down to the horizontal black line (Figure 3), and if the current tool is the crosshairs, it switches to the panning hand.  Though automatic mode switching is often to be avoided, in this case the mode (the current tool) is highly visible and users can fluidly begin panning after zooming, something they always did in our observations. The hand works as one would expect: by dragging on the page's surface, the user pulls the page in the direction she desires. The user can also pan the page using the joypad, the four-way thumb pad at the bottom of the iPaq. In casual testing, zoomed-in users would move the page beneath the target until the target was over a page element. Then they would zoom out and say, "oh, *that's* where that is" when they saw the element's location beneath the target in the context of the whole page.

### Rapid Serial Visual Presentation

In WebThumb, users see text as designers intended, in the same layout as it would appear on the desktop. For wide or tall paragraphs, however, this can make reading difficult when zoomed in because it requires panning.

Cognitive psychologists have studied rapid serial visual presentation (RSVP): the showing of text word-by-word in a fixed focal position [7]. Evidence shows as many as 12 words per second are readable in controlled situations. Reading speeds vary greatly depending on font, size, color, content, display medium, etc., but RSVP promises reading speeds within normal ranges.

WebThumb provides the RSVP tool  to support reading. With it, the user can tap on a block of text and have it "played" rapidly one word at a time (Figure 4). The text is shown in its original face but increased considerably in size to 15-point on the iPaq screen. As a result, text can be displayed even larger than in typical desktop settings, yet no more space is consumed than that needed for a single word.



**Figure 4.** RSVP "playing" text from the UIST '02 homepage. The current word, *Mutualité*, is shown in the center.

This technique trades time for space by reusing the same small set of pixels over time.

### Joypad and Hard Buttons

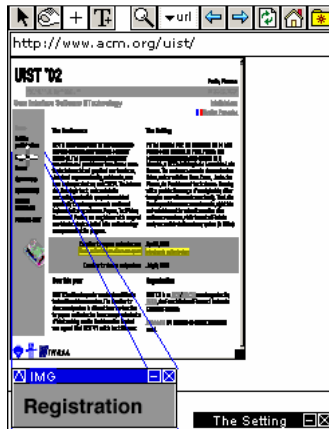
In determining a word's presentation duration, the RSVP engine takes into account the length of a word, whether it is capitalized or not, and the punctuation after it. Our design iterations indicate that in practice, good pacing can be achieved using a base duration of 150 ms and adding 25 ms per letter, 40 ms for a capitalized word, 310 ms for a comma, and 420 ms for a period. Miscellaneous punctuation also incurred a slowdown of 25 to 40 ms. In the end, users in informal testing were able to read paragraphs of unfamiliar text at a natural pace.

It is important to leverage handhelds' physical form when designing for them. WebThumb uses two physical features of the iPaq: the two hard buttons to the left of the joypad, and the joypad itself. The hard buttons to the left of the joypad can be used by the user's left-hand thumb to cycle through the tools. The right hand, holding the stylus, can then take action on the interface. This mimics the way people use their hands with physical objects: the non-dominant hand selects objects and the dominant hand takes action on those objects. Obviously the buttons would be reversed for left-handers.

When zoomed in, the joypad does what one might guess: it pans to expose more of the page in the direction pressed.

When zoomed out, the joypad can be used to pick up the elements in a page in a directional fashion from the current target point (Figure 5). This works by iterating over the thumbnail from the target to the thumbnail's edge, hit-testing for links and images at each pixel along the way. As soon as an element is found, it is picked up and shown.

An example will help illustrate this technique. In Figure 5, a user has placed the target atop a vertical list of navigation links at the left side of the page. When the user presses the down arrow on the joypad, the target moves to the center of the topmost link and picks it up. When the user presses down again, this link goes away and the link below it is picked up, and so on down the column. If the user sees a link he wants to follow, he can press in on the joypad (the joypad itself is a button) to follow the link. This is a fast and easy way to discover the contents of dense lists of links.



**Figure 5.** The joypad can be used to pick up elements in a cardinal direction from the target. Here, the user picks up each link in a column by pressing down on the joypad.

### Browsing Capabilities

WebThumb implements the familiar capabilities of browsers: back, forward, refresh, stop, and home. Also, when tapped, the URL button shows a text field into which any address can be entered (Figure 5). With this implementation, we were able to surf the Web in a way that allowed informal comparisons with current browsers (such as Pocket IE) during our iterative design. WebThumb seemed more satisfying for sites with lots of links, images, and dense content. Pocket IE was more satisfying for all-text pages where only some gradual scrolling was required. We want to formally study the two experiences as a part of future work.

### TECHNOLOGY

We implemented WebThumb as a desktop prototype in Microsoft Visual C++ using the MFC framework. We rendered thumbnails of pages by running Internet Explorer as a separate process, then capturing and shrinking IE's renderings to thumbnail size. This approach allowed us to get high quality rendering and full browsing in our prototype with a minimum of effort, allowing more resources to be devoted to exploring interaction techniques. To interact with and extract individual web page elements, we accessed IE's Document Object Model (DOM) through Microsoft's Component Object Model (COM). Traversal

of the DOM tree allowed us to isolate page elements and hit-test them, which was crucial for our interaction techniques.

### FUTURE WORK

Informal testing shows promise but more rigorous validation of our desktop prototype is required; in particular, comparisons with Pocket IE. Once complete, we intend to port WebThumb to an iPaq device. We also want to explore more sophisticated RSVP playback, and consider adding thumbnail enhancements [8,11].

### CONCLUSIONS

We believe WebThumb demonstrates that web browsing on handheld devices can and should depart from the experience of its desktop predecessor by using a variety of interaction techniques designed for the device. The techniques we employ allow for an enriched browsing experience different from a mere "shrunk desktop."

### ACKNOWLEDGEMENTS

The authors would like to thank Jeffrey Nichols, James Fogarty, and Dan Boyarski for their contributions. This work was supported in part by the National Science Foundation under grants IIS-0121560 and IIS-0117658. Any opinions, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

### REFERENCES

1. Buyukkokten, O., Garcia-Molina, H., Paepcke, A., Winograd, T. "Power browser: Efficient web browsing for PDAs." *Proc. CHI '00* (The Netherlands, April 2000), ACM Press, 430-437.
2. CMU Kinetic Design Group, examples available at [http://www.cmu.edu/cfa/design/kdg/kt/kt\\_examples/kt\\_kid.html](http://www.cmu.edu/cfa/design/kdg/kt/kt_examples/kt_kid.html)
3. Ford, S., Forlizzi, J., and Ishizaki, S. "Kinetic typography: Issues in time-based presentation of text." *CHI '97 Extended Abstracts*, (Atlanta, March 1997), ACM Press, 269-270.
4. Kamba, T., Elson, S.A., Harpold, T., Stamper, T., Sukaviriya, P. "Using small screen space more efficiently." *Proc. CHI '96* (Vancouver BC, April 1996), ACM Press, 383-390.
5. Kim, L., Albers, M.J. "Web design issues when searching for information in a small screen display." *Proc. SIGDOC '01* (Santa Fe, October 2001), ACM Press, 193-200.
6. Myers, B.A. "Using hand-held devices and PCs together." *Communications of the ACM* 44, 11 (November 2001), 34-41.
7. Potter, M.C. "Rapid serial visual presentation (RSVP): A method for studying language processing." In Kieras, D.E. and Just, M.A. (eds.), *New Methods in Reading Comprehension Research*. Lawrence Erlbaum, Hillsdale, NJ, 1984, 91-118.
8. Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., van Dantzich, M. "Data mountain: Using spatial memory for document management." *Proc. UIST '98* (San Francisco, November 1998), 153-162.
9. Small D., Ishizaki, S., and Cooper, M. "Typographic space." *CHI '94 Conference Companion* (Boston, April 1994), ACM Press, 437-438.
10. Wong, Y.Y. "Temporal typography." *CHI '96 Conference Companion* (Vancouver BC, April 1996), ACM Press, 408-409.
11. Woodruff, A., Faulring, A., Rosenholtz, R., Morrison, J., Pirolli, P. "Using thumbnails to search the web." *Proc. CHI '01* (Seattle, March-April 2001), ACM Press, 198-205.