SteeringWheel: A Locality-Preserving Magnification Interface for Low Vision Web Browsing

Syed Masum Billah[1] **Vikas Ashok**[1] **Donald E. Porter**[2] **IV Ramakrishnan**[1]

[1]Department of Computer Science, Stony Brook University, NY, USA

{sbillah, vganjiguntea, ram}@cs.stonybrook.edu

[2]Department of Computer Science, University of North Carolina at Chapel Hill, NC, USA

porter@cs.unc.edu

ABSTRACT

Low-vision users struggle to browse the web with screen magnifiers. Firstly, magnifiers occlude significant portions of the webpage, thereby making it cumbersome to get the webpage overview and quickly locate the desired content. Further, magnification causes loss of spatial locality and visual cues that commonly define semantic relationships in the page; reconstructing semantic relationships exclusively from narrow views dramatically increases the cognitive burden on the users. Secondly, low-vision users have widely varying needs requiring a range of interface customizations for different page sections; dynamic customization in extant magnifiers is disruptive to users' browsing. We present SteeringWheel, a magnification interface that leverages content semantics to preserve local context. In combination with a physical dial, supporting simple rotate and press gestures, users can quickly navigate different webpage sections, easily locate desired content, get a quick overview, and seamlessly customize the interface. A user study with 15 low-vision participants showed that their web-browsing efficiency improved by at least 20 percent with SteeringWheel compared to extant screen magnifiers.

Author Keywords

Accessibility, magnifier, low-vision, web-browsing, user-interface, locality, visual impairments.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces–*input devices and strategies*. K.4.2. Computers and Society: Social issues–*assistive technologies for persons with disabilities*.

INTRODUCTION

Low vision, a visual impairment that cannot be fully corrected even with glasses, medication or surgery, includes loss of peripheral or central vision, blurred vision, extreme light sensitivity, tunnel vision, & near-total blindness [1, 5]. People with low vision predominantly rely on screen magnifiers (e.g., ZoomText, MaGic and many others [2]) to interact with computer applications, including web browsers [42]; some low-vision users may also supplement magnifiers with screen readers (e.g., JAWS [31], NVDA [37]) that are mostly used by blind people to read aloud screen content.

Although screen magnifiers have been around for a while, their usability has not been well-researched [30]. A recent study, for instance, identified several problems faced by low vision users with current generation magnifiers for interacting with computing devices [42]. Firstly, the magnifiers indiscriminately magnify the screen content, including whitespace, which causes important local contextual information such as visual markups (e.g., borders) and semantic relationships between different UI elements (e.g., a checkbox and its label) to be occluded from the user's viewport. For example, in Figure 1, notice how the element labeled "From" is isolated from its neighboring local context "City or Airport" in the top-left piece. Similarly, in the bottom-left piece the user will have no idea that the form field in the magnification viewport is a drop-down menu. These kinds of isolations compel users to manually pan, with no restraints, over the occluded portions, and mentally stitch

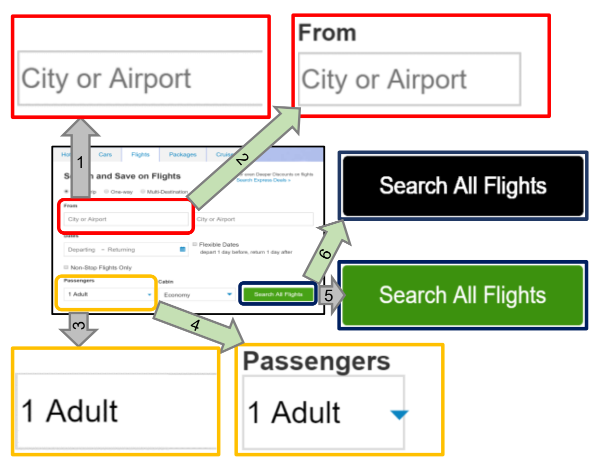

Figure 1. Magnification in ZoomText vs. SteeringWheel in a travel booking site where each box is a logical segment. Odd numbered arrows show the magnification of the corresponding logical segment in ZoomText; even numbered arrows show the same in SteeringWheel.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM 978-1-4503-5620-6/18/04...$15.00

https://doi.org/10.1145/3173574.3173594

together different fragments of the content by themselves, thereby increasing the cognitive burden. Panning and scrolling also makes it frustrating and time-consuming for the users to get a quick overview of the webpage and locate elements of interest; this effort can become considerably high for large magnification factors (e.g., zoom 16x).

Secondly, the heterogeneity of the visual presentation (e.g. fonts, colors, contrast, borders, etc.) of the different page elements such as images and forms, may necessitate frequent dynamic customization of the magnification interface as the user navigates a web page. However dynamic customization of the interface in extant magnifiers is tedious and often disrupts users' browsing flow [43]. For example, in ZoomText, the user has to open a separate configuration window, bring up a menu which may have nested sub-menus, choose an option, return to the previous browsing location to see the effect of the chosen option and repeat this tedious process until the desired customization is achieved.

In this paper, we present SteeringWheel, a *semantics-based locality-preserving magnification interface* to address the aforementioned limitations of current screen magnifiers. SteeringWheel incorporates knowledge about the semantics of different web UI elements and inter-element relationships rooted on the concept of a *Logical Segment* (LS). A LS is a collection of related UI elements sharing common spatial and functional properties with a perceivable visual boundary. In a sense LS intuitively captures local context. LS may be recursively partitioned into sub-LSs. For example, in Figure 1, the Flights tab and the corresponding Form is a LS, and the form fields are sub-LSs. Keeping related elements of an LS within the user's viewport, i.e., *locality preservation*, is achieved in SteeringWheel by adhering to the following key design principles: First, the focus of magnification is limited at any time to a single LS, and the cursor movement is confined (i.e. *cursor is clipped*) to the boundaries of this LS. Second, the white space in an LS is differentially magnified so that the non-white-space elements are kept as close to each other as possible. Third, certain elements within a segment are selectively rescaled post magnification, if necessary, so as to retain them in the user's viewport. An illustration of locality preservation is shown in Figure 1, where the width of the dropdown-list (yellow box) is reduced, and also the space between the list and its label ("Passengers") is reduced to keep the entire form field within the user's viewport.

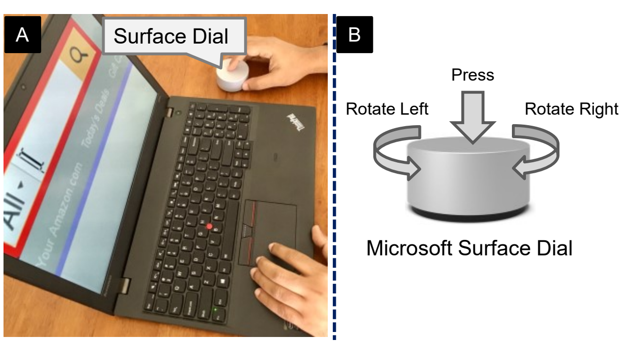

Figure 2. (A) A low vision user using SteeringWheel with Surface Dial. (B) The Surface Dial and its gestures.

SteeringWheel's magnification interface uniquely leverages a simple set of *rotate and press gestures with audio-haptic feedback* provided by a physical dial, an off-the-shelf Microsoft Surface Dial[1], for stepping through the LS hierarchy as well as all the elements within a LS (see Figure 2). These gestures, which can be done with ease and rapidity, make it possible to navigate the LS hierarchy to get a quick overview of the webpage and locate the content of interest. The reach of such gestures can be readily extended to all magnification related operations by embedding the various configuration options on the dial's built-in radial dashboard (see Figure 4.13), which eliminates the need for interacting with separate configuration windows, thereby avoiding diversion of attention from the current browsing location. From this dashboard, the user can select and tune any configuration option via simple gestures, and immediately observe the corresponding effect - à la WYSIWYG. To further speedup the customization process, the custom magnification settings of one LS is automatically applied to other similar semantic segments (e.g. search result items), thereby avoiding needless repetition. Support for customization, as incorporated in SteeringWheel, thus addresses the problem of seamless dynamic customization of the magnification interface. By uniquely combining content semantics, differential magnification, space reduction algorithm for preserving local context, cursor clipping to reduce panning effort, rotate and press gestures and WYSIWYG customization interface, SteeringWheel exemplifies a new generation screen magnifier.

A user study with 15 low-vision participants showed that their web-browsing efficiency improved by at least 20 percent with SteeringWheel compared to extant screen magnifiers such as ZoomText.

RELATED WORK

Screen Magnification. Accessibility problems with screen magnifiers have received attention from early on [7, 13, 14, 33]. Kline et al. [33], for instance, highlighted the locality issues introduced by the magnifier's viewport, proposed a dual-mode magnifier - mobile and anchored, and conducted a user study of this magnifier. In the mobile mode, the magnification window follows the cursor on the screen, whereas in the anchored mode, a fixed screen area is designated as the viewport, and the screen region surrounding the cursor, as it is moved around, is displayed in this viewport. The findings of their study revealed the need for discernible visual markers to indicate cursor location and configure the magnifier seamlessly as well. Cursor enhancements for low vision are also reported in Fraser et al.

[1] https://www.microsoft.com/en-us/store/d/surface-dial/925r551sktgn

[21], who reviewed the findings of prior research studies on assistive devices for low vision users.

Zhao et al. [50] conducted a large-scale usability study of screen magnifiers for older and visually impaired users. They made several recommendations based on this study including configuring the magnification mode to "full-screen" (i.e., setting the magnification viewport to cover the entire screen), by default. Several web accessibility issues with screen magnifiers (e.g., ZoomText) were also identified by Theofanos et al. [43]. These included the challenge of obtaining the gist of webpages under the constraint of limited viewport, and the disorientation caused by the scaling of empty white-space under magnification. They also made several recommendations for improving web accessibility for low vision users.

Christen et. al. [17] studied the effect of magnification and contrast on reading performance under a variety of simulated low vision conditions. Their key finding was that there's no "one-size-fits-all" accessibility solution for the spectrum of eye conditions that low-vision impairment entails. Interestingly, this finding was also reinforced by our own user study evaluating SteeringWheel. Seamless configuration of magnification interface is essential to address this finding.

Hallett et al. [26] studied the impact on reading comprehension with and without word wrapping. Screen magnifiers were found to cause discomfort because of their lack of support for word wrapping. Recently, expanding on an earlier work [41], Szpiro et al. [42] articulated several accessibility challenges with modern magnifiers that provided the context for this paper (see Introduction section).

Improving the usability of magnification has also been explored to some extent. For example, Widget Lens of Agarwal et al [3] does widget-specific magnification based on the widget's semantic properties. Bigham [9] proposed utilizing the redundant space in a screen to amplify the text without causing "negative side effects" such as the amplified text overlapping with other screen objects and other content in the neighborhood. A recent work [49] explores the use of special head-mounted displays for people with low vision.

In summary, existing works have focused on certain specific aspects of low-vision usability. SteeringWheel provides a more generic and comprehensive solution to these usability problems. Finally, we mention that the applicability of dial for blind users has been recently explored in [10].

Target Acquisition. It is important for people with low vision to be able to point and select targets on the screen with ease and efficiency. Towards this objective several approaches have been proposed [6, 12, 23, 28, 32, 46], the main idea being the modification of the presentation of targets and/or cursor. In contrast to these approaches that gauge improvements based on Fitts's law [36], Object Pointing [24] overrides the default cursor behavior to directly jump from one interface object to another, thereby bypassing the intervening white spaces altogether. Although it is an interesting assistive device, especially for people with motor impairments, overriding the default ad-hoc pointing behavior of the mouse is not desirable in practice. SteeringWheel also achieves the same effect as Object Pointing without altering the mouse behavior.

Overviews and Contexts. Having an overview of the web page and the surrounding context of the web elements is essential for locating and operating on contents of interest. The *overview+detail* interface is a specialized multi-window arrangement where one window provides the overview while the other window is used for displaying a detailed view of the selected content fragment [27]. But visual overviews do not serve low-vision users. In the *focus+context* technique, such as fisheye [25] and hyperbolic browser [35], users can view the selected content fragment without requiring a second window. However, the rest of the content is rendered opaque or distorted. This technique is usually used for viewing high resolution graphical data such as maps.

UI Adaptation. Adapting UI elements to meet an individual's accessibility requirements is another line of research. Supple [22], which exemplifies this paradigm, automatically personalizes an interface for people with visual and motor disabilities. However, it requires a formal description of the interface elements. As Bigham [9] points out, this is not practical for web pages, since JavaScript and CSS frameworks allow the creation of arbitrary web elements precluding standardization of their descriptions. Furthermore, the presence of many web page elements, makes the adaptation algorithm quite expensive in practice.

Webpage segmentation. A principal component driving the SteeringWheel interface is the hierarchical organization of Logical Segments. Semantic understanding of web pages is a classic research problem dating back to the inception of the Web (e.g., [18, 19, 20]). The use of web page semantics in accessibility is also a long-standing research topic (e.g., [29, 38]). The construction of the Logical Segments in SteeringWheel is based on well-established web-segmentation techniques (e.g., [4, 16, 48, 51]). Broadly speaking, these techniques identify segments in a web page at various levels of granularity, based on the observation that related items in a web page often exhibit consistency in visual presentation style and spatial locality on the page. The identified segments are organized into a semantic hierarchy. We note that Apple's VoiceOver screen-reader also employs a basic notion of semantics using Web spots—landmarks that separate different segments on the webpage. The hierarchy of semantic groups in SteeringWheel is a tree, similar in structure to the HTML DOM of the web page. This hierarchy forms an abstract semantic layer over the DOM tree.

Audio-haptic interfaces. Several research works have demonstrated the usability and applicability of audio-haptic interfaces for accessible web browsing [34, 39, 40, 45, 47]. SteeringWheel incorporates tactile feedback as navigational aids by sensing the boundaries of Logical Segments.

THE DESIGN OF STEERINGWHEEL

Design Guidelines

The following design guidelines were formulated after an extensive literature review, cited in the itemized guidelines.

G1. Bimanual or manual interaction.
Users should be able to interact with SteeringWheel either with their dominant hand alone or with both their hands as is done in [8]. This is useful for accommodating the needs of a wide range of visual disabilities.

G2. Clipped cursor and selective feedback.
Users' input (i.e., cursor movement) should be confined to the boundaries of a single area of interest at any time to reduce panning effort [42], and the output (i.e., audio, haptics) should only convey information pertaining to that area of interest.

G3. Preservation of webpage layout.
The spatial layout and arrangement of web elements should not be repositioned, as low-vision users do not want a "different" website [43]. However, minor adjustments such as *opportunistic accessibility improvement* [9] can be made to the web page.

G4. WYSIWYG (What You See Is What You Get) customization.
Users should be able to immediately see the changes on the screen in real time as they customize the interface [49]. The customization effort should be minimum, seamless and not disruptive to web browsing [33, 43].

G5. Content Personalization.
Users should be able to selectively apply different customizations to different semantic segments within a page [3, 26]. These customizations should be saved and automatically reapplied the next time the same webpage is visited to avoid repetition.

SteeringWheel System Description

The user interacts with SteeringWheel in two ways - gestures (e.g., rotate, press) and actions (e.g., point, click, scroll). The interpretation of user input at any time is dependent on the two operation modes of SteeringWheel at that instant, namely, *navigation* and *anchor*, with navigation being the default mode. In *navigation* mode, SteeringWheel applies a clear discernible (red dotted-line) border to the Logical Segment (LS) currently in focus, and reads aloud its contents. The user can shift focus to the next/previous LS with a simple rotate right/left gesture; SW will automatically apply the discernible dotted-line border to the next/previous LS and read aloud its content. For example, in Figure 4, notice how the focus shifts from "Search", "Advertisement", to "Product Items" logical segments when the user performs rotate right gestures.

To further explore a LS of interest and perform magnification operations on it, the user first needs to switch to the anchor mode. This is done by a simple *press* gesture after locating that LS in the navigation mode; SteeringWheel automatically locks the viewport, applies a solid red border to that LS, and confines the mouse cursor movement (i.e., cursor clipping) to that LS. In the *anchor* mode, the user can zoom in/out using *rotate* gestures, pan over the LS content using the cursor (if necessary), and use the *press & hold* gesture to bring up the Configuration Dashboard for customizing the magnification settings for this LS (more details appear later). Whenever the user performs a zoom operation, SteeringWheel automatically applies custom space-reduction technique to keep the content of the LS close to each other and preserve important visual cues such as borders and markers, i.e., preservation of local context.

Also, in the anchor mode, the user can navigate the sub-LSs of the LS one by one using an accelerate gesture; this gesture automatically switches the mode back to navigation. If there are no sub-LSs, then SteeringWheel interprets the *accelerate* gesture as a *point & click* operation. For example, if the current segment is an HTML button, *accelerate* gesture will simulate a *click* operation on that button, thus providing an alternative to clicking with a mouse cursor. Note that the *accelerate* gesture is only available in the *anchor* mode. In other words, the user can navigate down the semantic hierarchy only after entering the anchor mode at any segment. The rationale behind this design choice is to encourage users to consume information at top levels of the semantic hierarchy where the contextual information is higher, and therefore minimize the cognitive burden involved in mentally reconstructing semantic relationships between sub-segments; if the users can view the segment content comfortably after adjusting a few interface parameters such as zoom and contrast, there is no need to explore the sub-segments one-by-one separately and mentally piece together all the gathered information. Also note that the accelerate gesture is automatically triggered whenever the zoom level exceeds a very high threshold.

Further, at any time, irrespective of the operation mode, the user can switch SteeringWheel's focus back to the LS from any of its sub-LSs using the *double press* gesture. To provide such a navigational feature, SteeringWheel analyzes the webpage DOM when it is loaded in the browser, identifies and extracts the various LSs and their relationships (e.g., parent-child, sibling, ancestor, etc.), and constructs a hierarchy of these logical segments organized in an object-oriented fashion. The details regarding how this is accomplished, as well as the other core components of SteeringWheel are presented next.

Extracting LSs and constructing LS Hierarchy

SteeringWheel extended the well-known VIPS segmentation method [16] to identify the different segments on the page. This extension leveraged textual metadata (e.g., class) in DOM nodes for segmentation. Our segmentation algorithm with this extension showed 97% accuracy, making it ready for widespread use. Note that SteeringWheel is not tied to VIPS per se; any segmentation technique can be used in place of VIPS. The *root* segment in the LS hierarchy contains the entire webpage. The child segments under the *root*

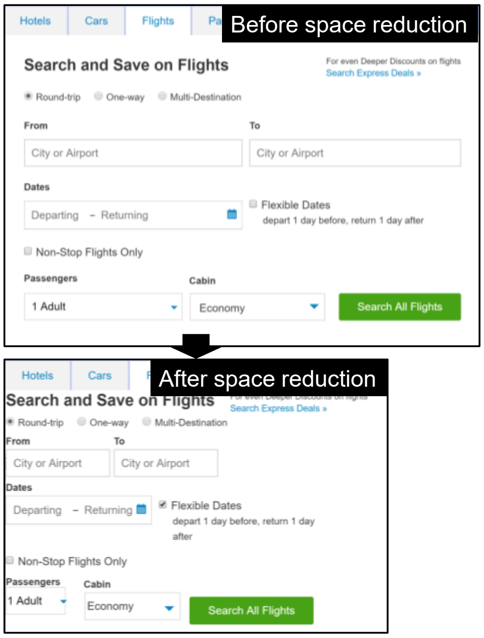

Figure 3. Illustration of space-reduction algorithm.

consist of different high-level visually-segregated blocks of the webpage such as menu, sidebar, search-box, main-content, footer, etc. Each of these segments can in turn have their own children which are segments themselves, and so on. By default, SteeringWheel starts in the *navigation* mode and the focus or viewport is set to the first child of root.

Preserving Local Context or Locality Preservation

SteeringWheel tries to preserve local contexts or locality via two ways: space reduction and cursor clipping.

Space Reduction: The goal of space reduction algorithm is to keep related elements of an LS together even after magnification. SteeringWheel's space reduction algorithm treats the target LS, say a form (see Figure 3) as a hierarchy of nodes consisting of sub-LSs (e.g., form fields) and their descendants (e.g., textboxes, labels). At any node, the space-reduction algorithm preemptively (i.e., prior to magnification) scales down various attribute values such as margin width, border dimensions, padding, etc., to ensure that the white space does not get indiscriminately magnified after zooming. This process is recursively applied to all the nodes in the hierarchy bottom-up starting at the leaf nodes; the space reduction at each node is propagated up the hierarchy and accumulated. Post-magnification overlaps between sub-LSs and other descendants, are avoided using techniques borrowed from Bigham et. al. [9]. Figure 3 illustrates the application of this process on an example form LS. Notice how the sizes of all the form field elements, the distances between them, as well as the overall size of the form itself have all been reduced.

Cursor Clipping: In the anchor mode, the cursor is clipped or confined to the selected segment, irrespective of the zoom factor. Therefore, users cannot move their cursor beyond the boundaries of the segment. Of course, these boundaries may be pushed off-screen due to magnification; however, users are still allowed to scroll and pan over the entire segment. This design choice of restricted panning allows SteeringWheel to avoid many problems of panning such as accidental cursor flick, i.e., accidentally moving the cursor out of the current LS.

Gesture	**Surface Dial**
press	Pressing the Dial
press-n-hold	Pressing and holding the Dial
rotate	Rotating the Dial
accelerate	Rotating the Dial fast
double press	Pressing the Dial twice in quick succession
triple press	Pressing the Dial thrice in quick succession

Table 1. Mapping of SteeringWheel gestures to Surface Dial.

Gestures

The selection of gestures was inspired by the findings of an earlier work [10] that demonstrated their effectiveness in hierarchical navigation of web content by blind users. SteeringWheel supports the following gestures, with Table 1 listing how SteeringWheel mapped these gestures to two devices: Mouse and Dial.

Rotation: Interpretation of this gesture is dependent on the mode of operation, as well as the selected wheel-dashboard option (if any).

Press: Pressing the *dial* toggles the SteeringWheel between the *navigation* and the *anchor* modes.

Acceleration: If the angular acceleration of the dial in a direction (i.e., left or right) exceeds a user specified threshold, then SteeringWheel treats it as an acceleration gesture. This gesture is only available in *anchor* mode and ends with a long "buzz" haptic feedback.

Double Press: An alternative to *accelerate left (right)* gesture for right (left)-handed user. This gesture is always available to users.

Triple Press: Starts or stops reading out the text content of the current segment.

Press-n-Hold: brings a wheel dashboard (see Figure 4.13). User can use rotate gestures to go through this menu and choose one of the options by performing a *press* gesture. In fact, dashboard menu contains options to simulate every other gesture, thereby providing an alternative.

Feedback in Navigation Mode.

On every *rotation* gesture, the *dial* provides a short "tick"-like haptic feedback and reads out a short title or description (e.g., "third item of search results") associated with the focused LS. Also, when the focus is on the last (first) LS at any level in the hierarchy of LSs, a rotate right (left) gesture produces a long "buzz"-like haptic feedback similar to that provided by the system in [10], to indicate to the users that they have reached the last (first) LS in that hierarchy level.

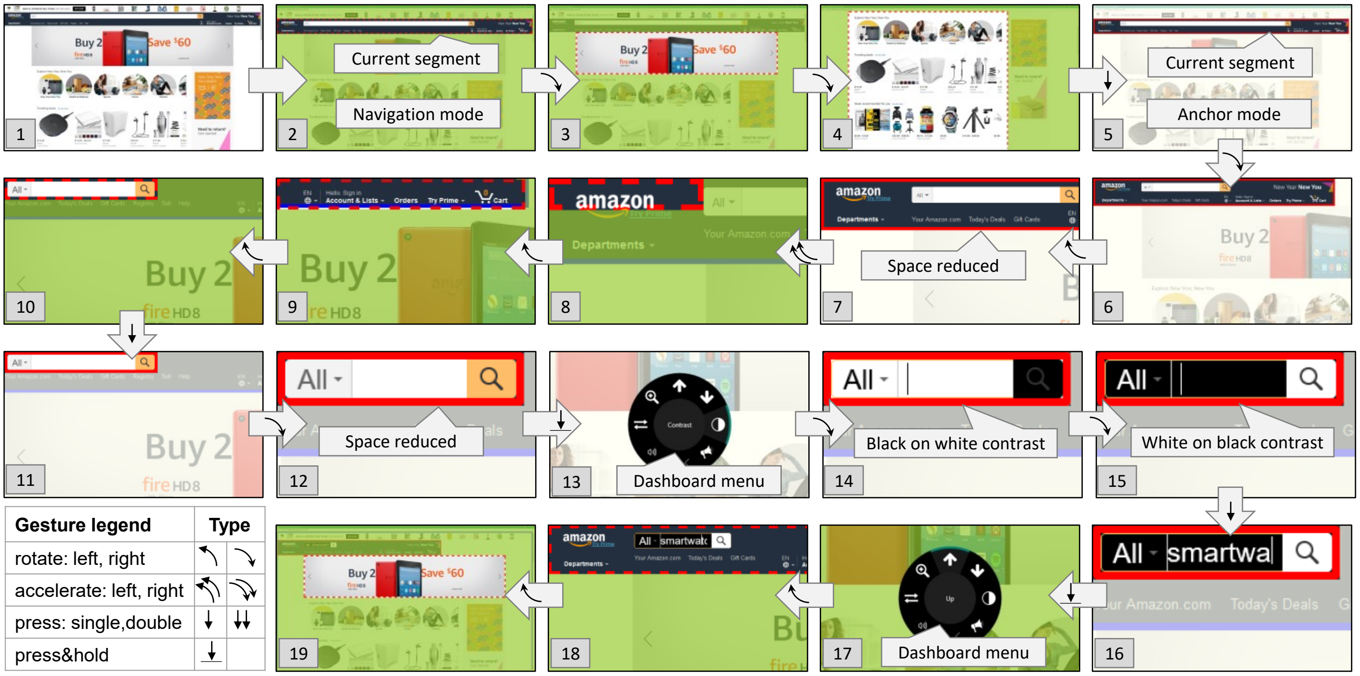

Figure 4. An illustration of SteeringWheel workflow: User is searching for "smartwatch" on the Amazon site. Thumbnails with green-shaded background correspond to navigation node, whereas the remaining thumbnails correspond to anchor mode. User starts from the top-left and performs a gesture (in the arrow) to go to next figure. The bottom-left figure shows the gesture legend.

Customizing Interface in Anchor Mode

Seamless non-disruptive customization of interface parameters for different LSs on the page is needed for low-vision users since many websites either fail to follow or only partially follow web accessibility guidelines for low vision browsing (e.g., [44]). Some customizations (e.g., cursor enhancement) are common to all segments, whereas others are more segment-specific.

The *press-n-hold* gesture brings up the configuration dashboard (see Figure 4.13). The dashboard holds several options such as zoom, contrast, brightness, color, etc., with zoom being the default focused option. To select any other option, the user can simply rotate the focus to that option and press to select that option. This overrides the default behavior of *rotate* gesture in the anchor mode to that selected option. As in the case of zooming, the user can see the effects in real time, when they perform the *rotate* gestures. For example, if the option selected is brightness, the user can instantly see the screen brightness increasing/decreasing as they do *rotate right*/*rotate left* gesture. Also, notice in Figure 4 (thumbnails 6 & 7) how the webpage is realigned after magnification such that the anchored LS does not go off-screen; SteeringWheel uses affine transformation to perform this realignment.

Illustration: Putting It All Together

Figure 4 is a user interation scenario illustrating how SteeringWheel is used on the Amazon website homepage. Figure 4.1 depicts the unmagnified website that is visible after the page is loaded. SteeringWheel now is in the default navigation mode. The user navigates to different LSs (i.e., menu with search form, advertisements, product items) by rotating (right or clockwise) the Dial (Figure 4.2-Figure 4.4). To avoid distraction and to highlight the current LS, in navigation mode, the foreground is obscured by semi transparent greenish-yellow shade except for the currently focussed LS. After navigating back and forth, the user now presses the Dial to anchor the SteeringWheel to the "Menu" LS that contains the "Search" form (Figure 4.5).

In the anchor mode, SteeringWheel applies a solid border to Menu and shades the rest of the screen with the semi-transparent beize color. Also, the cursor is locked/clipped within this Menu boundary. The user then rotates the Dial to magnify the contents of the Menu LS (Figure 4.6-Figure 4.7). Notice how the Menu is kept from moving off screen by realigning the webpage. Now, the user can either directly click on the search input box with mouse if (s)he can see it, or choose to explore the sub-LSs of Menu individually one-by-one. In Figure 4.8, the user performs the accelerate right gesture to move focus to the first sub-LS, i.e., the Amazon logo, and then continuosly performs the rotate right gesture (Figure 4.9-Figure 4.10) until the focus is on the "Search" LS. The user then anchors SteeringWheel to the Search LS (Figure 4.11) by pressing the dial once.

The user now magnifies the content of the Search LS (Figure 4.12) by rotating the Dial. Notice how space reduction is done on this LS, e.g., the reduced width of the search input textbox, so that the entire input-box is made clearly visible in the user's viewport. The user now decides to apply a custom high contrast theme to the input-box. To do that, the user brings up the configuration Dashboard (Figure 4.13) using a press & hold gesture. By rotating the Dial, the user can select different configuration options on this dashboard; in this scenario, the user picks the contrast option (position 6 clockwise) and then presses the dial to dismiss the dashboard. Rotating the Dial now applies different contrast theme one-by-one (e.g., white on black, black on white) in a WYSIWYG style to the Search LS (Figure 4.14-Figure 4.15). Now that the user is comfortable with the color contrast, (s)he finally enters the search term "smartwatch"in the input box (Figure 4.16). The user continues to explore and interact with other LSs on the webpage (Figure 4.17-Figure 4.19s).

EVALUATION

We conducted an IRB-approved user study to assess the effectiveness and usability of Steering-Wheel. Specifically, we aimed at validating the following hypotheses:

- **H1**: SteeringWheel has higher efficiency in completing browsing tasks involving continuous navigation within the page, compared to ZoomText.
- **H2**: With SteeringWheel, it takes lesser effort to locate, understand and interact with the desired segment, compared to ZoomText.
- **H3**: SteeringWheel has higher usability rating and user satisfaction than ZoomText.

Participants

We advertised for study volunteers through local mailing lists and social-network websites. After preliminary screening via phone interviews, we recruited 15 low-vision participants, and conducted the study at the Lighthouse Guild in NY city. The inclusion criteria included familiarity with web browsing and ZoomText as the preferred magnifier.

None of the participants had any kind of motor impairment. The participants varied in age from 26 to 70 (mean: 41.7, median: 35, SD: 14.4), gender (8 males, 7 females), and web-browsing experience with magnifiers (10 *adept*, 5 *beginners*). On average, the participants indicated they browsed the internet for 1.5 hours daily. Table 2 presents the participant demographics. All participants were aware of their diagnosed eye condition; however, some of them were not sure about their precise visual acuity.

Study Setup

We designed real-world browsing tasks to capture the real challenges low vision people experience when browsing the internet with screen magnifiers. These tasks were transactional in nature, requiring a sequence of steps spanning multiple web pages and multiple sections within a webpage. The websites for these tasks, were selected from 2 categories: flight reservation and online shopping. For flight reservation, we chose *Priceline* and *Expedia*, and for shopping, we chose *Amazon* and *eBay*. We deliberately picked a well-known Amazon website to test if SteeringWheel could outperform the current state-of-the-art even if participants had prior experience with a website. On each of these websites, the participants were asked to complete the following 3 tasks in a sequence:

- **T1**: Find the search form, fill-out the form with experimenter-provided data (e.g., flight-reservation details, product name, etc.) and hit the search button.
- **T2**: On the search-results page, find the search-result element, check if the number of search-results are more than 10 by counting, then find the filtering options (e.g., number of stops, layover duration, departure times, price, rating, vendor) for the search results, set certain experimenter-specified filters, and go back to search results and recount if there are 10 or more.
- **T3**: Find the search result that satisfies certain experimenter-provided criteria (e.g., cheapest airfare, shortest travel time, number of layovers during flight reservation).

The participants performed these 3 tasks under the following 2 conditions (one website in each category per condition):

- **ZoomText with optional Audio**: Participants could use ZoomText with keyboard and mouse, and could optionally use the audio output of ZoomText. This was the baseline condition.
- **SteeringWheel with Surface Dial**: Participants could use the proposed SteeringWheel magnification interface with a Surface Dial alone or in conjunction with the touchpad or a computer mouse.

To minimize the learning effect, we counterbalanced the ordering of websites and conditions, while ensuring that no two websites belonging to the same category (e.g., flight reservation) is assigned to the same condition. Additionally, the websites within each category differed considerably in appearance and richness in content. Prior to the study, participants were given sufficient instructions and time (~30 min) to familiarize themselves with both conditions. We chose the following 2 websites for practice: Kayak (flight reservation), and Walmart (online purchase). During practice, the participants were even allowed to customize SteeringWheel by defining their own gestures, change the existing gestures, etc. These customizations were retained while they performed the actual tasks later.

To avoid confounds, we visually inspected the LSs returned by the segmentation algorithm and manually fixed certain error segments before conducting the study. Furthermore, for a fair comparison between the two conditions, we disabled the browser's default magnification (Ctrl++) feature so that the magnification during the experiment could only be done either by ZoomText or by SteeringWheel.

The experiment was conducted on a 15.5-inch Win10 ThinkPad laptop with 2880x1620 resolution. It had Internet Explorer 11, ZoomText 11 and SteeringWheel installed. Each participant was allotted 20 minutes to complete each task. If a participant failed to complete any task within the stipulated time limit, that task was recorded as incomplete along with any observational notes, and the experimenter completed the task on participant's behalf if necessary (e.g., filling out the search form and submit), before letting the participant start the next task.

ID	Age/ Sex	Diagnosis C: Congenital, A: Adventitious	Visual Acuity L: Left, R: Right	Tools Used ZT: ZoomText, VO: VoiceOver, OM: Optelec Magnifier
P1	35/M	Retinitis pigmentosa (C)	L:20/200, R:0	OM, MagicPro, JAWS, ZT
P2	48/M	Macular Telangiectasia (A)	20/150	Windows Magnifier, ZoomText
P3	46/M	Congenital cataracts (C)	L:20/240, R:20/200	OM, Pocket magnifier, Audio book, NVDA, ZT
P4	64/F	Retinopathy of prematurity (C)	L:20/200, R:20/200	Hand magnifier, Telescopic lens, iPhone camera, ZT.
P5	62/F	Macular degeneration (C)	L:0, R:20/300	CCTV, Window 's Magnifier, ZT
P6	37/M	Congenital cataracts (C)	20/800	JAWS, NVDA, VO, Mac's Zoom, ZT
P7	34/M	Albinism (C)	L:20/200, R:20/400	NVDA, AppVision, GW-Micro, Large display, iPhone, ZT
P8	27/F	Myopia strabismus (C)	20/600	Magnifier, Narrator, iPhone camera, ZT
P9	29/F	Albinism (C)	20/240	OM, CCTV, Portable CCTV, Zoom, ZT
P10	70/M	Glaucoma (A)	Unknown	Magnifier, Narrator, Larger Key Caps, Telescopic lens, ZT
P11	33/F	ROP & Glaucoma (C)	L:20/200, R:20/400	Zoom, VO, JAWS, Handheld magnifier, ZT
P12	52/M	Optic atrophy (C)	20/800	JAWS, iPhone, ZT
P13	32/M	Nystagmus (C)	20/120	Telescopic lens, Magnifier, ZT
P14	26/F	Pathological Myopia (A)	20/200	JAWS, Magnifier, Phone camera, Large display, ZT
P15	31/F	Pathological Myopia (A)	20/280	Large display, Narrator, JAWS, Amazon Echo, ZT

Table 2. Demographic Information of the 15 Participants.

All conversations during the study were in English. The experimenter took notes during a session. All sessions were video recorded and coded post transcription. The participants were also encouraged to think aloud, and mention the problems as and when they encountered them during the study experiment.

Data Collection and Analysis

We analyzed the experimenter's notes and the recorded videos to measure the following metrics: (i) completion time; (ii) number of times each SteeringWheel gesture was used; (iii) magnification factor chosen for different page elements; (iv) number of times the participant sought moderator's help; (v) reasons for task completion failures, including SteeringWheel errors; (vi) number of times a participant failed to identify a LS in one attempt; and (vii) number of attempts made before successfully identifying a LS. We also recorded any unusual browsing behavior that delayed the completion of assigned tasks, such as repeatedly navigating over the same content, confusion due to form-validation errors, or failure to recognize the LSs. At the end of the experiment, the participants were asked to complete the standard System Usability Scale (SUS) questionnaire [15] and a set of custom-designed open-ended questions eliciting comments and suggestions for both experiment conditions.

Results

In this section, we present the analysis of the collected measurements and subjective feedback, as well as compare the participants' performance and web browsing experiences under the 2 different study conditions (i.e., ZoomText and SteeringWheel). We conducted paired t-test to determine if any differences in measures between these 2 conditions were statistically significant.

Completion time

We found significant effect of study conditions on completion time for tasks T1 ($t(14) = 19.09$, $p < .0001$), T2 ($t(14) = 5.79$, $p < .0001$), and T3 ($t(14) = 12.90$, $p < .0001$). The mean completion times for tasks T1, T2, and T3 under each condition are shown in Figure 5. For task T1, when using SteeringWheel, the mean completion time (148.53s) was reduced by 40% compared to that of the baseline ZoomText (243.47s). For tasks T2 and T3, these reductions were 23% and 20% respectively (SteeringWheel: mean=242.20s, ZoomText: mean=515.33s, SteeringWheel: mean=475.33s, ZoomText: mean=596.67s). A closer inspection of the collected data reveals the reasons why participants spent different amounts of time on different tasks, and why SteeringWheel took significantly less time than ZoomText for each of the tasks.

Task T1. T1 was designed to measure target acquisition effort. While using ZoomText, the participants who used an average zoom level between 8x-12x, could quickly locate the form & comfortably fill the form. However, those using a zoom factor between 15x-25x, spent considerable time searching for the form itself, let alone filling it. However, with SteeringWheel, the participants used the *rotate* gestures to quickly navigate and identify the form without having to manually pan over the entire page searching for it.

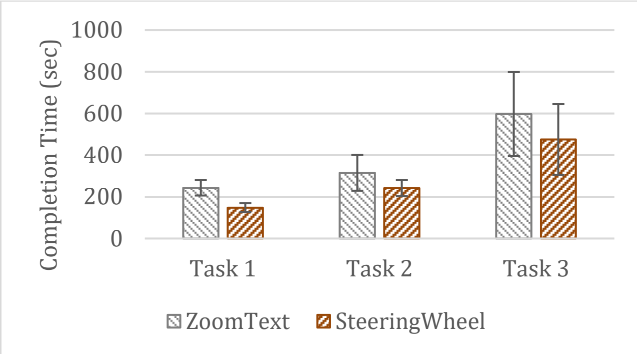

Figure 5. Completion times for 3 tasks T1, T2 and T3 using ZoomText and SteeringWheel. Error bars show ±1 SD.

Task T2. T2 was designed to measure how the magnifier affected participants' spatial understanding and spatial memory. While using ZoomText, the participants spent a lot of time searching for the *Search Results* and the *Filter Options* LSs. Notably, navigating to the Form Filters after going through the Search Results was found to be very cumbersome and disorienting, especially for participants requiring a very high zoom level (>20x). These participants failed to recognize the target even if it was in their direct line of sight. They navigated many times (5.6 times avg.) back-and-forth between the target and the surrounding context, before realizing that they had located the target. However, with SteeringWheel, the participants simply used the *double-press* gesture to navigate one level "up" the LS hierarchy, and then did a single rotate left gesture to move the focus to the *Filter Options* segment, while relying on the additional audio description provided by SteeringWheel.

Task T3. T3 was a visual-search task. While using ZoomText, the participants had trouble identifying the search-result item attributes (e.g., price, departure time), because they mostly had to view them in isolation. Sometimes, they unknowingly moved the focus to the attributes of a different search item and the realization and recovery from such errors further delayed the completion of this task. In the case of SteeringWheel, firstly, incorrect associations of data belonging to different search items never occurred as the cursor was clipped to an anchored LS. Furthermore, with ZoomText, users spent a lot of time repeatedly performing the same customization for every search item, especially on Amazon, where they had to choose different zoom levels for product descriptions and product images. With SteeringWheel however, they performed this customization once on the first search item, and these settings were automatically applied to all other items.

User Behavior and Subjective Feedback

Participants used different magnification factors ranging between 8x and 45x depending on their visual conditions. However, we observed that participants requiring similar zoom levels exhibited similar behavior. Based on this observation, we can partition the participants into 3 groups.

Group 1 (8x-12x). Participants (P2, P3, P4, P5, and P14) who used a magnification factor between 8x and 12x fall under this group. Each of these participants seldom explored the LS hierarchy, mostly confining their interaction to the top levels. In other words, they opted for panning instead of exploring sub-LSs. All participants in this group used only their dominant hand to interact with either the dial or the mouse; they did not use these two devices with both their hands simultaneously. However, a few preferred to use the dial in conjunction with the touch-pad instead of the mouse.

Group 2 (15x-25x). Participants (P7, P8, P9, P11, P13, and P15) in this group used a magnification factor between 15x to 25x. They preferred to use the dial with their dominant hand and the mouse with their other hand simultaneously. On being explicitly asked about their choice of moose + dial, they stated that the availability of dial reduced their dependence on the mouse.

Group 3 (30X+). Participants (P1, P6, P10, and P12) in this group used over 30x magnification factor and preferred the dial over the mouse. They stated that at a high magnification level, it was hard to follow the mouse-cursor movement and to understand the underlying page content. Some of these participants were screen-reader users, while others were in the process of learning a screen-reader.

Use of mouse. We noticed that at the beginning, participants scrolled the content using mouse-wheel. But as the study progressed, they started to scroll less frequently, and relied more on the dial to navigate the content.

SteeringWheel Interface Customization

The participants customized the gestures, visual presentation and internal threshold values (e.g., threshold for differentiating *rotate* gesture and *accelerate* gesture). 8 participants indicated that the accelerate gesture was a bit confusing. Instead, these participants opted for either an alternative wheel-dashboard option, or an automatic focus shift to the first sub-LS after reaching a threshold zoom level; the reason being that occasionally what was perceived by them as an accelerate gesture turned out to be a rotate gesture. The remaining 7 participants adjusted the threshold value for differentiating between rotate and accelerate gestures. Similarly, the participants avoided the triple press gesture by choosing the alternative dashboard option. Participants also customized different border styles and color according to their vision condition and preferences.

Usability Rating

At the end of each study session, every participant was administered the standard System Usability Scale (SUS) questionnaire where they rated positive and negative statements about each study condition on a Likert scale from 1 for strongly disagree to 5 for strongly agree, with 3 for neutral. There was a significant difference in the SUS score between ZoomText (mean = 69.40, SD = 4.09) and SteeringWheel (M = 80.8, SD = 3.38) conditions, $t(14) = -7.45$, $p < .001$. Participants stated that they preferred

SteeringWheel to ZoomText, as ZoomText was very cumbersome to use, especially for transactional tasks involving continuous navigation. All of the above results and observations validate our hypotheses H1, H2, and H3.

Post-Evaluation

Finally, we administered a brief, Likert-type questionnaire (1 means strongly unfavorable or very bad, 5 means-strongly favorable or very easy, and 3 means neutral) to solicit participants' opinions and suggestions regarding SteeringWheel, and a histogram summary of the feedback is presented in Table 3. As shown in Table 3, the participants gave positive feedback to almost all questions except Q6 where they expressed concerns about the *accelerate* gesture.

DISCUSSION

While most commercial and academic magnifiers such as ZoomText and Kline et al. [33] offer multiple modes of magnification (e.g., lens and full-screen), SteeringWheel only offers the full-screen mode. This design choice was influenced by the findings of an earlier research work [50] that showed that low-vision users primarily operated in full-screen mode. Having a single magnification mode also significantly simplifies the interface. However, note that unlike these current magnifiers, SteeringWheel allows magnification operations on a single LS at any given time (Design Guideline G2), thereby achieving an effect equivalent to placing a lens under that LS.

A unique aspect of SteeringWheel is that navigation and magnification both leverage the semantics of the content. This offers a lot of benefits including efficient navigation. For instance, the user need not manually move the cursor (with the accompanying lens) from one LS to another as in extant magnifiers; simple rotate gestures will automatically bring the adjacent LSs to focus, with the mouse cursor repositioned to the center of the focused LS. For example, in webpages where a lot of vertical scrolling (e.g., blog, search-results, product-review) is required, locating the menu on the top-left or top-right corner of the screen was found to be very difficult and tedious, because users had to repeatedly zoom-out of their current context, move the cursor in the anticipated direction, and then zoom back in, until they acquired the target. But with SteeringWheel, they could accomplish this task with only a few gestures.

SteeringWheel was designed to operate over any website including text-rich sites, e.g., news and social media. But participants preferred to listen rather than view such content. SteeringWheel is not limited to Web browsing, and can be extended for any desktop and mobile applications, as the notion of DOM exists for these applications. However, unlike HTML DOMs, application DOM cannot be easily mutated (i.e., node attribute values cannot be straightforwardly modified for space reduction). To address this, we can leverage our previous work on Sinter [11], where we generate a virtual user interface for an existing application, and then mutate the DOM of this interface.

Question \ Response (1 to 5)	1	2	3	4	5
1. How easy is it to rotate the Dial compared to panning with the mouse?	0	0	2	8	5
2. How easy is it to learn SteeringWheel?	1	2	0	4	8
3. How important is it to view the segment boundaries?	0	0	4	3	7
4. How useful is audio feedback?	0	1	6	6	2
5. How useful is haptic feedback?	0	0	0	6	9
6. How easy is it to perform gestures?	1	1	6	5	2
7. How easy is it to use Dial and Mouse together with both hands?	1	2	2	3	7
8. How easy is it to fill forms with SteeringWheel?	0	0	1	7	7
9. How easy is it to customize the interface with the Dashboard?	0	1	2	7	5
10. How noticeable were the effects of locality preservation?	0	0	1	9	5

Color Legends (5 color bins):

0-3	4-6	7-9	10-12	13-15

Table 3. Post-experiment questionnaire and responses. The columns labeled as 1 to 5 show how many times a particular response (1 to 5) was received for questions (in the rows).

SteeringWheel gestures can be implemented on other input devices such as gaming mouse and touchpads, and are not exclusive to the Surface Dial. For example, the rotate gesture can be mapped to the rotation of a mouse wheel and the press gesture could be mapped to the press of the mouse wheel.

Limitations of SteeringWheel: (a) it takes time to get used to hierarchical web browsing with gestures; (b) interacting with websites with lots of animation and video maybe problematic as space reduction may not be desirable; and (c) it takes time to get used to the two modes of SteeringWheel operation, and switch between the two modes without any confusion.

CONCLUSION AND FUTURE WORK

This paper presents the design, implementation and a user study of SteeringWheel, a locality-preserving magnification interface, using an off-the-shelf physical dial for low-vision web browsing. Our study findings indicate that the constructive synergy between the dial, the LS hierarchy of the webpage, and locality preservation in SteeringWheel, makes for a better user experience compared to extant screen magnifiers. In future, we plan to continue work on the space reduction algorithm for small-screen devices such as smartphones and release the algorithm as a browser plug-in so that users can use it as a lightweight magnifier.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful feedback. Swathi Sekar, Sai Rachana Patel, and Kavya Sivanesan contributed towards the system implementation. This research is supported by NSF: IIS-1447549, NEI/NIH: R01EY02662, NIDILRR: 90IF0117-01-00.

REFERENCES

1. AFB, Glossary of Eye Conditions. Retrieved from http://www.afb.org/info/living-with-vision-loss/eye-conditions/12#L.
2. AFB, 2017. Screen Magnification Systems. Retrieved from http://www.afb.org/prodBrowseCatResults.aspx?CatID=39.
3. Agarwal, B. and Stuerzlinger, W., 2013. Widgetlens: A System for Adaptive Content Magnification of Widgets. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*. British Computer Society, 1-10.
4. Álvarez, M., Pan, A., Raposo, J., Bellas, F., and Cacheda, F., 2010. Finding and Extracting Data Records from Web Pages. *Journal of Signal Processing Systems 59*, 1, 123-137.
5. AOA, Common Types of Low Vision. Retrieved from https://www.aoa.org/patients-and-public/caring-for-your-vision/low-vision/common-types-of-low-vision.
6. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A., 2003. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch-and Pen-Operated Systems. In *Proceedings of the of INTERACT*. 57-64.
7. Bederson, B. B. and Hollan, J. D., 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*. ACM, 17-26.
8. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D., 1993. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 73-80.
9. Bigham, J. P., 2014. Making the Web Easier to See with Opportunistic Accessibility Improvement. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 117-122.
10. Billah, S. M., Ashok, V., Porter, D. E., and Ramakrishnan, I. V., 2017. Speed-Dial: A Surrogate Mouse for Non-Visual Web Browsing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 110-119.
11. Billah, S. M., Porter, D. E., and Ramakrishnan, I. V., 2016. Sinter: Low-Bandwidth Remote Access for the Visually-Impaired. In *Proceedings of the Eleventh European Conference on Computer Systems*. ACM, 1-16.
12. Blanch, R., Guiard, Y., and Beaudouin-Lafon, M., 2004. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 519-526.
13. Blenkhorn, P. and Evans, D. G., 2006. A Screen Magnifier Using "High Level" Implementation Techniques. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 14*, 4, 501-504.
14. Blenkhorn, P., Evans, D. G., and Baude, A., 2002. Full-Screen Magnification for Windows Using Directx Overlays. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 10*, 4, 225-231.
15. Brooke, J., 1996. Sus-a Quick and Dirty Usability Scale. *Usability evaluation in industry 189*, 194.
16. Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y., 2004. *Vips: A Vision Based Page Segmentation Algorithm.* Microsoft technical report.
17. Christen, M. and Abegg, M., 2017. The Effect of Magnification and Contrast on Reading Performance in Different Types of Simulated Low Vision. *2017 10*, 2 (2017-05-16).
18. Chung, C. Y., Gertz, M., and Sundaresan, N., 2002. Reverse Engineering for Web Data: From Visual to Semantic Structures. In *Proceedings 18th International Conference on Data Engineering*. IEEE, 53-63.
19. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., and Tomlin, J. A., 2003. Semtag and Seeker: Bootstrapping the Semantic Web Via Automated Semantic Annotation. In *Proceedings of the 12th international conference on World Wide Web*. ACM, 178-186.
20. Fensel, D., Decker, S., Erdmann, M., and Studer, R., 1998. Ontobroker: Or How to Enable Intelligent Access to the Www. In *Proceedings of of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*. CiteSeer.
21. Fraser, J. and Gutwin, C., 2000. A Framework of Assistive Pointers for Low Vision Users. In *Proceedings of the fourth international ACM conference on Assistive technologies*. ACM, 9-16.
22. Gajos, K. Z., Wobbrock, J. O., and Weld, D. S., 2007. Automatically Generating User Interfaces Adapted to Users' Motor and Vision Capabilities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 231-240.
23. Grossman, T. and Balakrishnan, R., 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 281-290.
24. Guiard, Y., Blanch, R., and Beaudouin-Lafon, M., 2004. Object Pointing: A Complement to Bitmap Pointing in Guis. In *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 9-16.

25. Gutwin, C., 2002. Improving Focus Targeting in Interactive Fisheye Views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 267-274.
26. Hallett, E. C., Dick, W., Jewett, T., and Vu, K.-P. L., 2018. How Screen Magnification with and without Word-Wrapping Affects the User Experience of Adults with Low Vision. In *Advances in Usability and User Experience: Proceedings of the Ahfe 2017 International Conference on Usability and User Experience, July 17-21, 2017, the Westin Bonaventure Hotel, Los Angeles, California, USA*, T. AHRAM and C. FALCÃO Eds. Springer International Publishing, Cham, 665-674.
27. Hornbæk, K. and Frøkjær, E., 2001. Reading of Electronic Documents: The Usability of Linear, Fisheye, and Overview+Detail Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 293-300.
28. Hourcade, J. P., Nguyen, C. M., Perry, K. B., and Denburg, N. L., 2010. Pointassist for Older Adults: Analyzing Sub-Movement Characteristics to Aid in Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1115-1124.
29. Huang, A. W. and Sundaresan, N., 2000. A Semantic Transcoding System to Adapt Web Services for Users with Disabilities. In *Proceedings of the 4th International ACM Conference on Assistive Technologies*. ACM.
30. Jacko, J. A. and Sears, A., 1998. Designing Interfaces for an Overlooked User Group: Considering the Visual Profiles of Partially Sighted Users. In *Proceedings of the third international ACM conference on Assistive technologies*. ACM, 75-77.
31. JAWS, 2013. Screen Reader from Freedom Scientific. Retrieved from http://www.freedomscientific.com/products/fs/jaws-product-page.asp.
32. Kabbash, P. and Buxton, W. A. S., 1995. The "Prince" Technique: Fitts' Law and Selection Using Area Cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., 273-279.
33. Kline, R. L. and Glinert, E. P., 1995. Improving Gui Accessibility for People with Low Vision. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., 114-121.
34. Kuber, R., Yu, W., and McAllister, G., 2007. Towards Developing Assistive Haptic Feedback for Visually Impaired Internet Users. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1525-1534.
35. Lamping, J., Rao, R., and Pirolli, P., 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., 401-408.
36. MacKenzie, I. S., 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-computer interaction 7*, 1, 91-139.
37. NVDA, 2013. Nonvisual Desktop Access. Retrieved from http://www.nvda-project.org/.
38. Ramakrishnan, I. V., Stent, A., and Yang, G. L., 2004. Hearsay: Enabling Audio Browsing on Hypertext Content. In *International World Wide Web Conference (WWW)*.
39. Rotard, M., Knödler, S., and Ertl, T., 2005. A Tactile Web Browser for the Visually Disabled. In *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*. ACM, 15-22.
40. Soviak, A., Borodin, A., Ashok, V., Borodin, Y., Puzis, Y., and Ramakrishnan, I., 2016. Tactile Accessibility: Does Anyone Need a Haptic Glove? In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 101-109.
41. Szpiro, S., Zhao, Y., and Azenkot, S., 2016. Finding a Store, Searching for a Product: A Study of Daily Challenges of Low Vision People. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 61-72.
42. Szpiro, S. F. A., Hashash, S., Zhao, Y., and Azenkot, S., 2016. How People with Low Vision Access Computing Devices: Understanding Challenges and Opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 171-180.
43. Theofanos, M. F. and Redish, J., 2005. Helping Low-Vision and Other Users with Web Sites That Meet Their Needs: Is One Site for All Feasible? *Technical Communication 52*, 1 (//), 9-20.
44. W3C, 2017. Accessibility Requirements for People with Low Vision. Retrieved from https://w3c.github.io/low-vision-a11y-tf/requirements.html.
45. Wong, E. J., Yap, K. M., Alexander, J., and Karnik, A., 2015. Habos: Towards a Platform of Haptic-Audio Based Online Shopping for the Visually Impaired. In *Open Systems (ICOS), 2015 IEEE Confernece on*. IEEE, 62-67.
46. Worden, A., Walker, N., Bharat, K., and Hudson, S., 1997. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 266-271.
47. Yu, W., Kuber, R., Murphy, E., Strain, P., and McAllister, G., 2006. A Novel Multimodal Interface for Improving Visually Impaired People's Web Accessibility. *Virtual Reality 9*, 2-3, 133-148.

48. Zhai, Y. and Liu, B., 2005. Web Data Extraction Based on Partial Tree Alignment. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 76-85.
49. Zhao, Y., Szpiro, S., and Azenkot, S., 2015. Foresee: A Customizable Head-Mounted Vision Enhancement System for People with Low Vision. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. ACM, 239-249.
50. Zhao, Z., Rau, P.-L. P., Zhang, T., and Salvendy, G., 2009. Visual Search-Based Design and Evaluation of Screen Magnifiers for Older and Visually Impaired Users. *Int. J. Hum.-Comput. Stud. 67*, 8, 663-675.
51. Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., and Ma, W.-Y., 2006. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 494-503.