Bringing Things Closer: Enhancing Low-Vision Interaction Experience with Office Productivity Applications

HAE-NA LEE, Stony Brook University, USA
VIKAS ASHOK, Old Dominion University, USA
IV RAMAKRISHNAN, Stony Brook University, USA

Many people with low vision rely on screen-magnifier assistive technology to interact with productivity applications such as word processors, spreadsheets, and presentation software. Despite the importance of these applications, little is known about their usability with respect to low-vision screen-magnifier users. To fill this knowledge gap, we conducted a usability study with 10 low-vision participants having different eye conditions. In this study, we observed that most usability issues were predominantly due to high spatial separation between main edit area and command ribbons on the screen, as well as the wide span grid-layout of command ribbons; these two GUI aspects did not gel with the screen-magnifier interface due to lack of instantaneous WYSIWYG (What You See Is What You Get) feedback after applying commands, given that the participants could only view a portion of the screen at any time. Informed by the study findings, we developed MagPro, an augmentation to productivity applications, which significantly improves usability by not only bringing application commands as close as possible to the user's current viewport focus, but also enabling easy and straightforward exploration of these commands using simple mouse actions. A user study with nine participants revealed that MagPro significantly reduced the time and workload to do routine command-access tasks, compared to using the state-of-the-art screen magnifier.

CCS Concepts: • **Human-centered computing → Accessibility**; *Empirical studies in HCI*.

Additional Key Words and Phrases: Office Productivity Software; Word Processor; Accessibility; Usability; Screen Magnifier; Low Vision

ACM Reference Format:
Hae-Na Lee, Vikas Ashok, and IV Ramakrishnan. 2021. Bringing Things Closer: Enhancing Low-Vision Interaction Experience with Office Productivity Applications. *Proc. ACM Hum.-Comput. Interact.* 5, EICS, Article 197 (June 2021), 18 pages. https://doi.org/10.1145/3457144

1 INTRODUCTION

Office productivity applications, such as word processors, spreadsheets, and presentation software are indispensable tools of modern society. To interact with these applications, people with low vision typically rely on a screen magnifier, an assistive technology that enables them to enlarge original screen content, and also pan the magnified content (i.e., move the magnifier focus) using special keyboard shortcuts and mouse actions. However, given that only a portion of the screen is visible at any instant with a magnifier, interaction with productivity applications can be very challenging, since the UI elements in these applications are spread out over the entire screen, e.g.,

Authors' addresses: Hae-Na Lee, Stony Brook University, Stony Brook, NY, 11794, USA, haenalee@cs.stonybrook.edu; Vikas Ashok, Old Dominion University, Norfolk, VA, 23529, USA, vganjigu@odu.edu; IV Ramakrishnan, Stony Brook University, Stony Brook, NY, 11794, USA, ram@cs.stonybrook.edu.

2573-0142/2021/6-ART197 $15.00
https://doi.org/10.1145/3457144

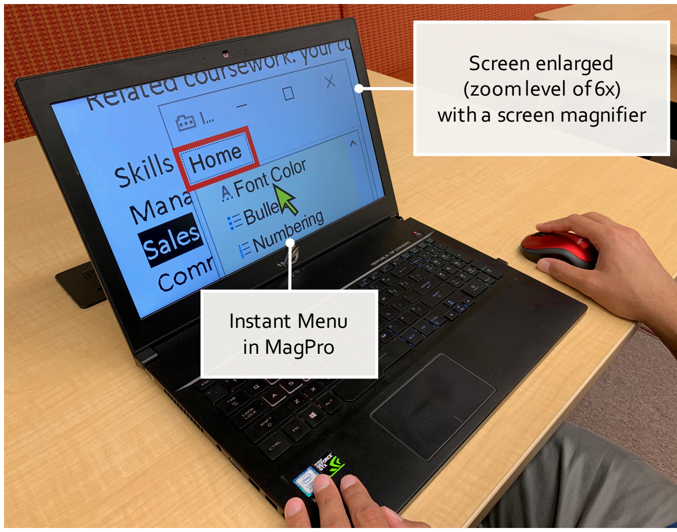


Fig. 1. A low-vision user accesses the desired command with a screen magnifier and MagPro. The user focuses on just one fragment of the screen and scrolls over the commands in MagPro's Instant Menu.

in Microsoft Word, text area is in the middle, upper ribbon and lower ribbon command menus are at the top, and so on. Unfortunately, the usability issues faced by low-vision screen-magnifier users while interacting with productivity tools remain as an understudied research topic [34].

To understand how low-vision users interact with productivity applications using a screen magnifier, and identify their usability issues and needs, we conducted a usability study with low-vision participants, where we observed them performing representative tasks on Office productivity software. From the study, we made many observations. The most notable of these was that the participants spent significant amount of time and effort panning back-and-forth and constantly adjusting the zoom level while applying formatting commands (e.g., `Font`, `Font Color`), which can be attributed to the absence of instant visual feedback when using screen magnifiers. Also, almost all participants only knew a few basic keyboard shortcuts, such as bold, italic, and underline, that constitute a very small fraction of the entire set of ribbon commands. The participants also had to repeatedly adjust the magnifier viewport while typing to keep track of the keyboard cursor and verify the entered text. In summary, these observations along with other study insights clearly demonstrated the need for additional interface support in screen magnifiers to facilitate more usable interaction with productivity applications. To fulfill this need, we designed and developed *MagPro* that augments the application interface in a way that enables low-vision users to quickly and easily perform important routine tasks such as accessing the application commands and confirming the effects of applying these commands.

The fundamental idea underlying MagPro is to push or deliver all the application commands within close proximity to the user's present magnifier focus in the edit area, preferably within the user's current viewport, so as to facilitate WYSIWYG (What You See Is What You Get) feedback, instead of the user having to switch context and search for these commands by panning and zooming. MagPro realizes this idea by augmenting the application interface with an additional *Instant Menu* that is accessible with a pointing device such as a mouse (see Figure 1). The Instant Menu contains all the application commands arranged in the form of a scrollable list. Furthermore, the Instant Menu is displayed within close proximity to the current user focus in the edit area, and is also compact, thereby it not only offers the user an instant feedback to the best extent possible, but also lets them focus only on a small fragment of the screen to view all the commands, which in turn reduces their panning, zooming, and even their eye-movement efforts. Furthermore, MagPro automatically adjusts the magnifier viewport in response to the movement of the keyboard cursor, so that the user can always see what they are typing, without having to readjust their magnifier focus constantly while typing. A user study with low-vision participants revealed that with the MagPro, the time to access desired commands as well as the perceived workload were significantly reduced, compared to using only their preferred screen magnifier.

We summarize our contributions as follows:

- Findings of a usability study with 10 low-vision participants that expose critical usability issues faced by screen-magnifier users when they interact with Office applications
- MagPro – An augmentation to the productivity application interface that mitigates usability issues by significantly reducing the amount of panning and zooming actions for performing routine tasks such as accessing commands and confirming command effects
- Findings of a user study with nine low-vision participants evaluating the effectiveness and potential of MagPro in improving the user experience with screen magnifiers while interacting with Office productivity tools

The rest of the paper is organized as follows. Section 2 describes existing literature that closely relates to our research. The findings of a qualitative study uncovering the usability issues of low-vision users are then presented in Section 3. Next, the detailed description of the MagPro architecture is provided in Section 4. An evaluation of MagPro in a user study with nine low-vision screen-magnifier users is then presented in Section 5, followed by a discussion in Section 6 on limitations of MagPro and future work. Finally, the concluding remarks are provided in Section 7.

2 RELATED WORK

Our work relates to existing research on: *(i)* usability issues faced by people having visual impairments, and *(ii)* techniques for improving usability of applications for visually-impaired users.

2.1 Usability Issues of People with Visual Impairments

While there exists extensive literature exploring the usability issues of blind screen-reader users [5, 7, 17, 18, 22], usability of screen-magnifier assistive technology [13, 25, 32] for people with low vision has been largely understudied [14, 27, 34, 38]. For instance, Jacko et al. [14] and Szpiro et al. [34] investigated low-vision user behavior, interaction strategies, and challenges of low-vision screen-magnifier users. Specifically, Jacko et al. [14] analyzed the mouse cursor movements made by low-vision users who had age-related macular degeneration to observe the interaction strategies of these users. In their experiment, they found that the size of icons is a significant factor that influences user performance; the larger the icon size, the better the performance. They also observed that both the cursor movement time and velocity were correlated to the severity of visual impairment. Specifically, the movement time significantly increased whereas the movement velocity significantly decreased, with reduction in visual acuity. Similarly, Szpiro et al. [34] conducted a study to understand the behavior of low-vision users when they interact with a variety of computing devices such as smartphones, tablets, laptops, and desktop computers. In their study, they observed that low-vision users rely on multiple assistive technologies (e.g., screen magnifiers and screen readers) simultaneously to do their everyday tasks on computing devices. However, they also found that despite using multiple assistive technologies, the low-vision users did not have adequate support to easily perform everyday computing tasks, thereby causing the interaction experience of these users to be inefficient and error-prone. They also observed that users needed to perform numerous gestures to comfortably view content, and constantly struggled while switching between applications or websites. Another key finding of their study was that the low-vision users preferred using their residual functional vision over text-to-speech to do their tasks.

While the above studies focused only on generic usability aspects of low-vision interaction with computing devices, Moreno et al. [27] focus on domain-specific usability issues faced by low-vision users while browsing the web. In their study, they found that only half of the participants could locate the specified target on a website. Additionally, they found that the participants confronted context-loss issues while trying to scroll the page horizontally. Specifically, they had to first move the magnifier viewport down the screen to focus on the horizontal scrollbar, then click and drag the scrollbar to the right, and finally pan the viewport back up to the main content of the page. During the study, the participants also complained about the inadequacy of visual contrast, lengthy lines of text, inflexible page layout, and absence of skip links. While a few of these domain-specific observations (e.g., scrolling) from this study may carry over to the scenario of productivity applications, the usability issues faced by low-vision users in these applications are still largely unclear, and therefore we uncover and address the productivity application-specific issues in this paper.

2.2 Usability Enhancement Techniques for People with Visual Impairments

The extant techniques for enhancing the usability of interaction with productivity applications have all focused on blind users who rely on text-to-speech screen readers [9, 19, 26, 28]. To the best of our knowledge, there are no extant techniques directly addressing the productivity application-specific usability needs of low-vision screen-magnifier users. Most of the extant low-vision usability solutions for computing applications have either focused on generic GUI enhancements [10, 11, 15, 21], or have mainly targeted web browsing [3, 4], smartphone applications [16, 31], and recently virtual and augmented reality [33, 36, 37].

The generic usability-enhancement techniques for low vision [11, 15, 21, 29] focus on modifying the existing GUI of applications or screen magnifiers. For instance, one of the seminal efforts in low-vision accessibility by Kline et al. [15], presents a set of accessibility tools that let users selectively magnify a portion of the screen area and also enable them to keep track of the mouse cursor location. Gajos et al. [11], on the other hand, propose personalization techniques that automatically generate GUIs that are tailored to the custom needs of users based on their motor and vision abilities. Specifically, they formulate GUI generation as a constraint optimization problem that searches for the optimal assignment of widgets to the elements in a given interface specification, while satisfying the constraints capturing user preferences and requirements. Similarly, Rothfux et al. [29] focus on generating low-vision friendly webpage interfaces by combining automated design-time generation of webpages with a state-of-the-art responsive design tool such as Bootstrap[1]. While these efforts designed for general-purpose interaction are indeed useful for improving low-vision usability in productivity applications, they are not sufficient to capture the domain-specific low-vision interaction needs associated with the productivity tools. Specifically, these techniques cannot address the usability issues caused by the distributed layout of content and controls – the main source of tedium and frustration for low-vision screen-magnifier users.

The domain-specific solutions for computer applications, have mostly centered around web browsing [1, 3, 4], where users primarily consume content rather than on their creation as in case of productivity tools. Bigham et al. [3] developed a magnification system that automatically figures out how much to enlarge webpage content without introducing adverse side effects, such as additional horizontal scrolling. Their fundamental idea is that webpages have a lot of empty space

[1] https://getbootstrap.com/

that can traded for enlarged content. Similarly, Billah et al. [4] present a context-preserving screen magnifier based on space compaction that attempts to keep related web elements close to each other within the magnified viewport, by discriminately magnifying the space around these elements. Furthermore, they provided a easy-to-use Dial input device for low-vision users to quickly navigate the content using simple rotate and press gestures. While these space-compaction techniques indeed help mitigate usability issues arising due to loss of local context when screen content is magnified, they are still incapable of handling scenarios that require users to simultaneously view different application segments (e.g., document content and control ribbons) that are spatially situated far from each other in the application GUI. In such scenarios, space compaction alone is unlikely to bring the desired segments close to each other in the magnified viewport. Our proposed solution MagPro was designed specifically to address these scenarios, via a custom-designed Instant Menu. The features realizing the vision of bringing the command and content within close visual proximity to each other in the magnifier viewport are unique to MagPro, and to the best of our knowledge, they are not supported by contemporary screen magnifiers.

3 UNCOVERING USABILITY ISSUES

We conducted a qualitative study with 10 low-vision users to examine and understand the interaction issues they face while interacting with productivity applications.

3.1 Participants

We recruited 10 screen-magnifier users (5 female, 5 male), with an average age of 47.8 (Median = 48.5, SD = 12.8, Range = 27-68). All participants were familiar with ZoomText screen magnifier, and they frequently used Microsoft Office tools at school, work, and home. The visual acuity of the participants ranged between 20/100 and 20/500. The participants were also aware of their diagnosis, which included a wide range of low-vision conditions such as glaucoma, retinitis pigmentosa, macular degeneration, diabetic retinopathy, corneal dystrophy, cone-rod dystrophy, and cataract. Also, all participants requested the speech narration feature of screen magnifiers to be turned on while doing the tasks. None of the participants had any motor disabilities. Table 1 presents the participants' demographic information.

3.2 Apparatus

The study was conducted using an Acer laptop running Windows 10, with ZoomText and Office 2016 suite installed. A traditional external keyboard and mouse were plugged into the laptop.

3.3 Design

The participants were asked to do the following tasks using the ZoomText screen magnifier:

- Create a one-page curriculum vitae in Microsoft Word.
- Create a one-page report describing a city in Microsoft Word.
- Create and fill up a timesheet in Microsoft Excel.
- Create and fill up a trip-expense report in Microsoft Excel.
- Create a Microsoft PowerPoint presentation about a school.
- Create a Microsoft PowerPoint presentation about a tourist place.

To avoid confounds, the content and commands for each task were controlled. Specifically, the experimenter explicitly specified the content to be typed and the commands to be accessed while the participants performed each task during the study. However, the experimenter did not specify the exact command option (e.g., font size value) to chose; the choice was left to the participants.

ID	Age/ Gender	Diagnosis (C - Congenital, A - Adventitious)	Visual Acuity		Max Zoom	Usage Frequency of Productivity Tools
			Left Eye	Right Eye		
P1	58/M	Retinitis pigmentosa (A)	0	20/500	8×	5 days/week
P2	49/M	Diabetic retinopathy (A)	Unknown	Unknown	5×	7 days/week
P3	55/F	Cataract (A)	20/200	20/100	4×	3-4 days/week
P4	62/M	Macular degeneration (A)	20/400	20/400	6×	5 days/week
P5	31/F	Corneal dystrophy (C)	Unknown	Unknown	5×	7 days/week
P6	45/F	Glaucoma (A)	20/400	20/200	5×	5 days/week
P7	48/M	Retinitis pigmentosa (A)	20/400	0	8×	2-3 days/week
P8	27/F	Glaucoma (C)	20/100	20/100	4×	5 days/week
P9	68/F	Macular degeneration (A)	Unknown	Unknown	6×	1-2 days/week
P10	35/M	Cone-rod dystrophy (C)	20/200	20/400	6×	2-3 days/week

Table 1. Participant demographics for the preliminary study. All information was self-reported by participants.

We adopted a concurrent think-aloud protocol. The ordering of tasks for each participant was counterbalanced using the Latin square method [6].

3.4 Procedure

The experimenter began the study by letting the participants adjust the magnifier settings such as zoom level, color inversion, and cursor appearance. Then the experimenter asked the participants to do the tasks in the predetermined counterbalanced order. The entire study was recorded (both screen and audio) with the permission of participants. The experimenter also took observation notes on the participants' interaction strategies, usability issues, and spoken utterances while they performed the tasks.

3.5 Data Analysis

The participants' transcribed utterances along with the experimenter notes were qualitatively analyzed using an open coding technique [30], where we iteratively went over the collected data, and identified key insights, pain points, and themes that reoccurred in the data.

3.6 Findings

The most notable findings that were common across all applications considered in the study are detailed next.

Difficulties in keeping track of current keyboard focus. All participants had to frequently readjust the magnifier focus to view what they were typing in the edit area, since the magnifier focus is only tied to the mouse cursor, but not the keyboard cursor. After some time, eight participants simply gave up readjusting focus, and relied only on text-to-speech audio to listen to what they were typing. Due to this, these participants on most occasions could not see their current context, i.e., the content that they were editing. Therefore, in the absence of visual feedback, six participants sometimes failed to capture grammar and spelling mistakes.

Frequent to-and-fro panning with constant zoom adjustment. To apply the commands and check the corresponding effects, all participants had to pan the magnifier viewport back-and-forth between their current keyboard focus in the content area and the commands in the ribbon area. For example, we observed that when choosing a line-spacing option for the text in their curriculum vitae, eight participants repeatedly moved the magnifier focus to-and-fro between the highlighted text and the line-spacing option menu, before settling in on a value that matched their presentation needs. Furthermore, during this process, the participants constantly had to adjust the zoom as the size of command icons and line-spacing options were different from the size of the text in the content area. Also, the participants could not see their current context in the content area, and therefore the context menu shown after a right click, was of little help as they could not instantly confirm the effects of applying a command; instead they had to first find their current keyboard focus on the screen before using the context menu.

Difficulties in horizontally panning over the ribbon commands. All participants explicitly indicated that they faced problems trying to horizontally navigate the ribbons while trying to locate a desired command. They also indicated that the GUI of ribbons made interaction laborious with a screen-magnifier, as they had to pan back-and-forth between the small-font list of ribbon names (e.g., `Home`, `Insert`, `Design`) located on the top-felt of GUI, and the ribbon commands that span the entire width of the screen. They mentioned that due to this design, it was not easy to switch ribbons, after panning over the commands from left-to-right within a given ribbon, as they had to now pan back to the list of ribbon names to select a different ribbon. Six participants stated that interacting with the ribbons was especially stressful because they had to move their head and eye gaze a lot, while constantly being vigilant about not unintentionally missing the desired command during panning.

Redundant and random panning actions while searching for a desired command option in a grid layout. All participants except P8 had to pan the magnifier viewport to view the entire set of command options (e.g., `Styles`, `Font Color`, `Design`) that was by default arranged in the form of a two-dimensional grid in the application GUI. Only two participants navigated the grid sequentially in a top-to-bottom left-to-right fashion, thereby not missing any option. The remaining eight participants navigated the grid in random directions while searching for the desired option. All these eight participants, on at least one occasion, struggled to locate the desired option and instead kept panning over the same options multiple times, thereby causing them to explicitly express frustration to the experimenter.

Difficulties in interpreting command icons. Six participants at least on one occasion, struggled to locate certain commands in the ribbon, even when the icons for those commands were within the magnifier viewport, because the corresponding tooltips describing the commands appeared outside the viewport. At the end of the study, these participants stated that they would rather prefer textual labels instead of icons for the application commands.

Summary. The findings of our study illuminate critical deficiencies of screen magnifiers as well as misalliances between screen magnifiers and application GUIs that are root causes of several usability issues that arise when low-vision users interact with Office productivity tools. All these identified usability issues contribute to extraneous interaction burden for low-vision screen-magnifier users in the form of two undesirable consequences – excessive panning/refocusing, and constant zoom readjustment. To address these issues, we designed MagPro explained next.

4 APPROACH

In the formative study uncovering usability issues, the participants expressed a desire for a command GUI that was within close proximity to their current screen-magnifier context, so as to facilitate almost instantaneous WYSIWYG (What You See Is What You Get) feedback. Furthermore, they wanted the commands in this GUI to be represented as a linear list that they can easily scroll over the commands with simple mouse actions without having to search by moving their eye gaze all over the magnifier viewport, i.e., two-dimensional space. These requirements informed our design choices for MagPro.

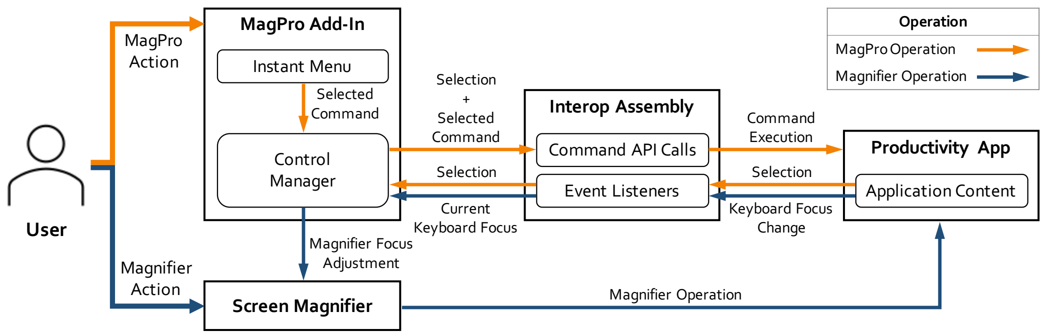


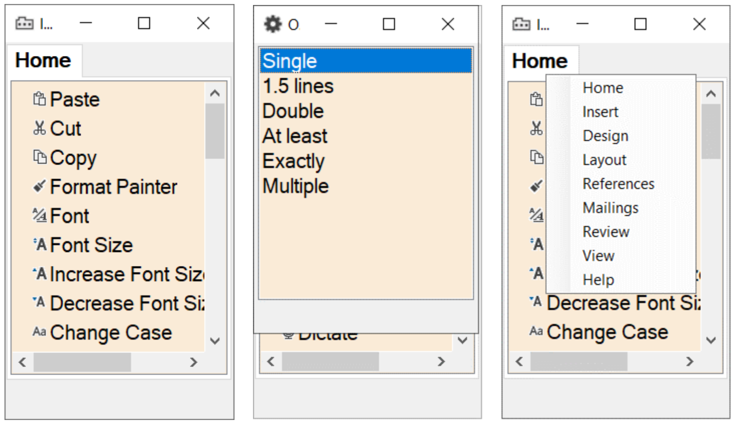
Fig. 2. An architectural overview of MagPro.

4.1 MagPro Design

MagPro reduces panning, zooming, and refocusing efforts by *(i)* providing a compact pop-up menu called *Instant Menu* that is displayed very close to a user's current context (i.e., selected text), and *(ii)* synchronizing the magnifier viewport location with the keyboard cursor to enable users to view what they are typing without having to manually readjust the magnifier focus. Details regarding these are provided next.

Overview. Figure 2 presents an architectural schematic of MagPro. MagPro was implemented as a Visual Studio Tools for Office (VSTO) Add-in [23]. As shown in Figure 2, MagPro has two core components, namely the *Instant Menu* and the *Control Manager*. The user interacts with MagPro via the Instant Menu using a pointing device such as a mouse. The Instant Menu (see Figure 3) contains all the application ribbon commands that are compactly arranged as a collection of one-dimensional scrollable lists in contrast to the two-dimensional grid-like arrangement of commands and their options in the original ribbon GUI of the productivity applications. On the other hand, the Control Manager is responsible for: *(i)* the execution of commands selected by the user in the Instant Menu; and *(ii)* tracking the *current keyboard focus* in the main edit area and automatically *adjusting the magnifier focus*, so that the user can always see what she typed in the edit area. MagPro add-in relies on the extant Microsoft Primary Interop Assembly [24] to obtain the *current context* (i.e., current keyboard focus and *selected text*) as well as automate command *executions*. Specifically, the Interop Assembly provides a wide range of *Command API Calls* and *Event Listeners* that support access and manipulation of application metadata, such as accessing the current selection in the main edit area (i.e., selected text), getting the current keyboard focus, setting the format characteristics of selected text, inserting objects, and accessing/setting the revisions data (e.g., `Track Changes`). Therefore, whenever the user selects a particular command in the Instant Menu, MagPro calls the corresponding method in the Interop Assembly to execute the command, thereby achieving the same effect as that observed when the user manually clicks the corresponding command in the application ribbon.

Note that the Instant Menu of MagPro is different from the standard context menu displayed in response to a right click in the following two aspects. Firstly, the context menu only offers a small set of predetermined commands. While these commands are indeed useful in many situations, they cannot replace the ribbons, i.e, they are not sufficient to complete many application tasks. Furthermore, the implementation of context menu is inconsistent across platforms, e.g., the context menu on macOS platform does not show a small grid of formatting commands that is shown in the one on Windows platform. Secondly, the context menu in applications is implemented as a "flyout" menu which is not suitable for low-vision screen-magnifier interaction, as the users will need to



(a) (b) (c)

Fig. 3. MagPro's Instant Menu: (a) list of commands for the default `Home` ribbon, (b) command options for `Line and Paragraph Spacing` command overlaying the Instant Menu, and (c) list of ribbons overlaying the Instant Menu.

pan the content back-and-forth to navigate the menu hierarchy, thereby resulting in loss of context. The MagPro's Instant Menu was therefore designed to overcome these limitations.

Instant Menu. Figure 3 shows the Instant Menu GUI of MagPro. A screen-magnifier user can access the Instant Menu with a simple mouse middle click action. To reprogram a mouse for supporting interaction with Instant Menu, we used the open-source `GlobalMouseKeyHook` library [20]. By default, the Home ribbon commands are arranged in the form of a scrollable list (see Figure 3a). Informed by the findings of the earlier usability study, MagPro shows both a command label and its corresponding icon side-by-side so that the user can easily identify each command. Also, all commands and their options are presented as lists, even if they appear as grids (e.g., `Font Color`) in the original ribbons. Furthermore, the size of each command label and icon are fixed to be equal in the Instant Menu to avoid zoom-level adjustments while interacting with it.

To navigate the list of commands in a ribbon, a user simply needs to scroll with a mouse wheel, and the commands will be brought into the viewport. This way, the user does not need to move her head or eye gaze while searching for the desired commands. From this list, the user can choose and apply a command on selected text or object using a left click. If a command has options (e.g., options for the `Line and Paragraph Spacing` command in Figure 3b), the option list will be shown as an overlay on top of Instant Menu so as to prevent panning and refocusing. The user can either scroll through this list and select an option, or simply close the list of options and return to the list of commands. To access other ribbons, the user can right-click on the ribbon tab (see Figure 3c) and select the desired ribbon with a left click. MagPro will then refresh the list of commands with those in the selected ribbon.

Execution of commands. As explained earlier, selection of a command in the Instant Menu causes MagPro to trigger the corresponding Interop API call (e.g., *selection.Font.Size = 12*), thereby simulating the effect of a mouse click on the actual command icon in the application ribbon. While some of the commands are applicable to the entire active main edit area (e.g., `Track Changes`, `View Comments`, `Accept All Changes`, `Margins`), other commands are mostly applicable to the current selection or focus in the main edit area (e.g., `Line Spacing`, `Bullets`, `Styles`, `Insert Table/Figure`). Therefore, the Control Manager of MagPro houses an ontology containing the method signatures and execution instructions for all the application commands. For example, in case of a command that is specific to the selected text (e.g., `Font Color`), the Control Manager fetches the current selection object using the Interop Assembly, and then sets the value of the appropriate property of this object (e.g., *selection.Font.ColorIndex = Word.WdColorIndex.wdBlue*). As the set of commands across different productivity applications are finite with large overlaps, scalability is not an issue for MagPro.

Real-time auto-adjustment of viewport. In addition to Instant Menu, MagPro keeps track of the user's current focus in the main edit area, and continuously updates the location of the magnifier viewport such that the user can always view what is being entered in the edit area. To achieve this, MagPro leverages the fact that the magnifier viewport is tied to the mouse pointer, and therefore makes the mouse pointer automatically follow the keyboard cursor during edits.

Illustration. Consider a user scenario where a screen-magnifier user is editing a document in Microsoft Word, and wants to add a WordArt style to a certain text. In order to do this task with a screen magnifier, the user has to first select that text and then move the magnifier lens from the current viewport showing the text to the top of the screen containing ribbons. Then in the `Insert` ribbon, the user pans to find the `WordArt` command and selects one of the options in its drop-down grid layout. The user then pans back to the selected text in the main edit area to check the resulting effect. If the resulting effect is not satisfactory, the user repeats this process all over again, once for each option of the `WordArt` command. This repeated panning back-and-forth between the text content and the ribbons makes the interaction experience tedious and cumbersome.

To do the same task with MagPro, instead of panning to the ribbons, the user simply executes a middle click on the mouse wheel to bring up the Instant Menu that is displayed right next to the selected text (see Figure 1). The user then right-clicks on the `Home` ribbon to access the list of other ribbons (see Figure 3c), and then left-clicks on the `Insert` ribbon that contains the desired `WordArt` command. MagPro then refreshes the contents of the Instant Menu with the commands of the `Insert` ribbon. The user then scrolls over to the `WordArt` command and left-clicks to select it, which in turn refreshes the Instant Menu with a list of `WordArt` command options. As these options are shown right next to the selected text, the user can instantly observe the effects of choosing any option, and therefore the user can quickly explore and settle for the option that best matches the user's needs.

5 EVALUATION

We performed a user study with low-vision screen-magnifier users to assess the MagPro.

5.1 Participants

We were able to recruit nine low-vision screen-magnifier users, and their demographics are shown in Table 2. The participants varied in age between 29-65 (Average = 47, Median= 47), and the gender representation was approximately equal (4 female, 5 male). As in case of earlier study, all participants indicated that they were familiar with Windows Magnifier and ZoomText screen magnifiers, and that they used Office applications frequently. Only two participants, P1 and P2, requested color inversion. None of the participants had any motor disabilities. As in case of preliminary study, all participants requested the speech narration feature of screen magnifier to be turned on during the study. To ensure external validity, we ensured that there was no overlap between the participants groups in the two studies.

ID	Age/ Gender	Diagnosis (C - Congenital, A - Adventitious)	Visual Acuity		Max Zoom	Other Productivity Tools
			Left Eye	Right Eye		
P1	51/F	Leber congenital amaurosis (C)	20/100	20/200	5×	Docs, Outlook
P2	65/M	Glaucoma (A)	0	20/100	4×	Outlook
P3	34/M	Optic atrophy (A)	20/100	20/200	4×	Docs, Sheets, Visual Studio
P4	47/F	Chorioretinal scarring (C)	20/400	20/200	6×	None
P5	37/M	Stevens-Johnson syndrome (A)	20/200	20/400	8×	None
P6	62/F	Glaucoma (A)	20/400	0	5×	Outlook
P7	29/F	Optic atrophy, retinitis pigmentosa (A)	20/700	0	8×	None
P8	53/M	Diabetic retinopathy (A)	20/200	20/400	6×	Outlook
P9	45/M	Albinism, nystagmus (C)	20/200	20/400	3×	Visual Studio, RStudio, SPSS

Table 2. Participant demographics for MagPro evaluation. Other productivity tools refer to software used by the participants in addition to Microsoft Word, Excel, and PowerPoint.

5.2 Apparatus

The user study was conducted using an Acer laptop running Windows 10 home edition, with ZoomText and Office 2016 suite installed. A traditional external keyboard and mouse were plugged into the laptop. All participants indicated that they were familiar with the Windows 10 home edition platform.

5.3 Design

With Microsoft Word as a representative productivity tool, given its importance and high popularity [2, 35], we asked the participants to do the following tasks, which were similar to the ones in the preliminary study:

- Create an article describing 'deserts'.
- Create an article describing 'rainforests'.

In a within-subject experimental setup, the participants were asked to do the tasks under two conditions, with one task per condition:

- Using ZoomText screen magnifier (baseline)
- Using the proposed MagPro

As in case of preliminary study, to avoid confounds, for each task, we used pre-specified content and controls for all participants. Specifically, the experimenter specified the contents to be typed as well as the commands to be accessed and applied to the content, and the participants performed the tasks themselves. To ensure fair comparison between conditions, the document lengths were set to be equal (1 page containing 3 paragraphs), and the participants were also asked to access the same number of controls in each condition (20 controls). Note, however, that there was no overlap in the controls chosen in the two conditions, so as to avoid learning effect. Also, the assignment of tasks to conditions and ordering of conditions were counterbalanced using the Latin square method [6].

5.4 Procedure

The conduction of the study was similar to that of the preliminary study. The experimenter began the study by allowing the participants to configure the magnifier settings, such as zoom level and color inversion, according to their preferences. Then the participants were asked to do the tasks in the predetermined counterbalanced order. For each task, the experimenter specified the content to be typed and the commands (e.g., formatting, review, design, insertions) to be accessed and executed. The entire study was recorded (both screen and audio) with the participant's permission. The experimenter also recorded observations regarding interaction strategies and usability issues.

5.5 Data Collection and Analysis

During the study, we collected the following metrics and data:

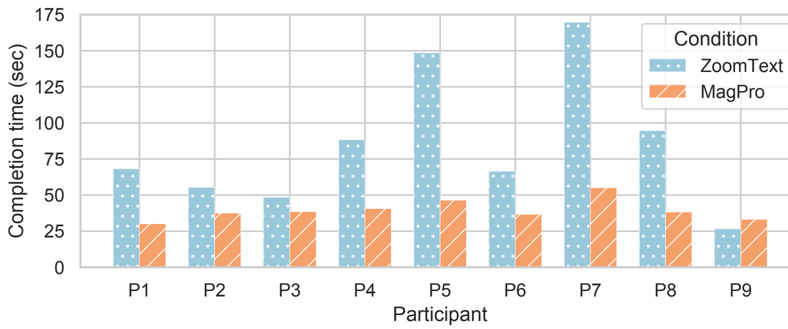
- Time taken by participants to find and execute a specified command
- Time taken by participants to confirm the effects of applying a command, i.e., focus back on the highlighted selection on which that command was applied
- System Usability Scale (SUS) score [8] assessing perceived usability
- NASA Task Load Index (NASA-TLX) score [12] capturing the perceived user effort
- Qualitative feedback from the participants as well as experimenter observations

Note that the time to find and execute a specified command was measured from the point user finished highlighting the specified text in the main edit area to the point the user selected the command in either the application ribbon or the MagPro's Instant Menu. Likewise, the time for checking command effects was measured from the point they selected the command on the ribbon or the Instant Menu until the point they verbally notified the experimenter after they verified the effects of command execution by panning back to the selected text on which the command was applied. In either case, the typing time was not considered in order to avoid confounds. The qualitative data from participants and experimenter notes were analyzed using grounding theory, specifically the open coding technique [30], where we iteratively went over the collected data, and identified key insights, pain points, and themes that reoccurred in the collected study data.

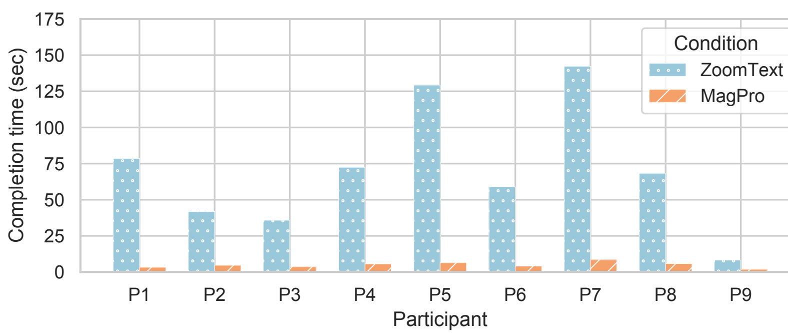
5.6 Results

Task completion times. Figure 4a shows the average time taken by each participant to find and apply the specified commands, and Figure 4b presents the average time that each participant spent confirming the effects of executing commands on the highlighted selection (e.g., text, shape). From these results, it is clear that MagPro helped in significantly reducing the task completion times for almost all (everyone except P9) participants, even though their visual conditions were different. However, P9 was slightly faster with screen magnifier while finding specified commands because of relatively better visual condition that had a wider field of vision and required only 3× zoom level, which enabled him to view a larger portion of the original screen content. For the same reason, P9 was also able to almost instantly check the command effects. Overall, the reductions in average task completion times achieved by MagPro for both finding/executing commands and confirming the effects of commands, were found to be statistically significant (Wilcoxon signed-rank test, $W = 44$, $z = 2.66$, $p = 0.007$ for finding commands; and $W = 45$, $z = 2.88$, $p = 0.003$ for confirming effects of command executions).

A closer analysis of the screen recordings and experimenter observations revealed that in the baseline ZoomText condition, the participants struggled to locate the specified command in the application ribbons, similar to the observations made in the preliminary study. Also, all participants could not experience the WYSIWYG (What You See Is What You Get) feedback when executing commands, i.e., they could not instantly check the effects of applying specified commands (e.g., `Font Color`), and therefore they spent considerable time panning to-and-fro between the main edit area with the highlighted text and the application ribbons on top of the screen containing the commands. Also, during this to-and-fro panning, all participants, on multiple occasions, struggled to find and refocus the magnifier lens back to the highlighted text from the ribbons after selecting a command or a command option, thereby further increasing the panning time while performing the task. On the other hand, in the MagPro study condition, none of the participants faced any



(a) Average time per participant for accessing commands.



(b) Average time per participant for checking command effects.

Fig. 4. Average task completion time taken by each participant.

issues in locating the specified commands, as they simply scrolled over the commands one-by-one in the Instant Menu while keeping their gaze fixed on one portion of the screen. With MagPro, five participants (P1, P2, P3, P6, and P9) experienced WYSIWYG feedback as both the highlighted text and the adjacent Instant Menu could fit into the magnifier viewport even after zooming the screen content. Although the remaining four participants could not enjoy the WYSIWYG feedback given the high zoom level, they did not pan much to confirm the effects of applying specified commands, as the Instant Menu was placed right next to the highlighted text on the screen, i.e., the participants panned a little right on the screen to apply the specified commands and then a back a little left to confirm the effects of command executions. This reduced panning effort with MagPro significantly decreased the task completion times compared to those in the baseline screen-magnifier condition.

Usability and perceived effort. To measure usability, we relied on the standard System Usability Scale (SUS) questionnaire [8] that comprises ten 5-scale Likert statements, where 5 - strongly agree, 1 - strongly disagree, and 3 - neutral. The responses to individual statements are then combined to generate a single score between 0 and 100, with higher scores indicating better usability. The overall average SUS score for MagPro as rated by the participants was significantly higher ($\mu = 86.11$, $\sigma = 4.58$) than that for the default ZoomText screen magnifier ($\mu = 66.67$, $\sigma = 13.69$). A paired t-test (as data was normally distributed) revealed that this difference in usability ratings between the two study conditions was statistically significant ($t = 4.355$, $p = 0.002$).

To measure perceived effort, we administered NASA Task Load Index (NASA-TLX) questionnaire [12], like SUS, generates a score between 0 to 100. However, lower TLX scores represent reduced task workload or effort, thereby indicating better results. In our study, the TLX scores were significantly lower for MagPro ($\mu = 17.55$, $\sigma = 6.34$) compared to that for ZoomText ($\mu = 58.03$, $\sigma = 8.13$). Furthermore, this difference in scores was found to be statistically significant (paired t-test, $t = -11.69$, $p < 0.00001$). Analysis of the individual responses revealed that for the ZoomText study condition, the Mental Demand, Temporal Performance, Effort, and Frustration sub-scales of the TLX questionnaire received significantly higher scores than the Physical Demand and Performance sub-scales, with Frustration receiving the highest weighted score. On the other hand, in the MagPro condition, all sub-scales received low scores, and we did not observe any patterns when comparing individual sub-scale scores.

By analyzing experimenter notes, we found that while accessing commands with the baseline ZoomText, almost all participants (except P9) struggled to get back to their previous location in the main edit area after panning away from it in order to find and apply a command from the application ribbons at the top of the screen. Furthermore, these participants on at least one occasion repeated the process of panning back-and-forth when certain chosen command option (e.g., color in `Font Color` command) did not match the command option specified in the task description. Also, four participants (P4, P5, P7, and P8) who used high zoom levels even had difficulties finding the commands in the ribbons as the tooltips of various command icons were shown outside the magnification viewport. All these factors made the interaction process tedious and cumbersome, which explains the low SUS and TLX ratings given by participants for baseline study condition.

Qualitative feedback. Analysis of the exit interview feedback revealed many recurring themes. For instance, all nine participants explicitly stated that it was easier and faster to access and apply commands using the MagPro compared to using only the ZoomText screen magnifier. As quoted by P2:

> *With this system, I don't have to go looking for stuff. All I have to do is click, and the controls appear right next to my current focus. After that I don't even need to move my head. I can simply scroll and go through the controls one-by-one without missing anything.*

With ZoomText screen magnifier, however, all participants except P9 mentioned that it was strenuous and frustrating to maintain their concentration while panning back-and-forth between their current context in the main edit area and the commands in the top application ribbons. The following quote by P6 best expressed this sentiment:

> *Every time I have to leave my current focus in order to access and apply a command, I subconsciously feel irritated, as I know this process is going to a long circus.*

Similar to the observations of the preliminary study, a majority (6) of participants indicated that it was mentally taxing to horizontally pan over the ribbons in the ZoomText study condition, as they had to pay extra attention in order to not accidentally miss the specified commands. These participants also indicated that it was harder to switch ribbons. With MagPro, however, all participants agreed that it was easier to switch ribbons.

Six participants preferred the text labels for representing commands over the image icons in the application ribbons. The remaining three participants stated that they preferred a combination of both text label and icon in the Instant Menu. No participant preferred just icons over the text labels for representing commands. Also, all nine participants stated that they found the list-style Instant Menu of MagPro easier to navigate than the grid-style GUI of default application ribbons. This was especially the case for participants with tunnel vision (P2, P6, and P7) who mentioned that they preferred keeping their mouse cursor fixed at one screen location and scrolling over the 1D list of commands as opposed to moving the mouse cursor all over the wide 2D ribbon of commands. Even P9, who had nystagmus condition that causes uncontrolled repetitive eye movements, stated that he preferred MagPro, even though he was faster with ZoomText, because it was less stressful to find desired commands in a list compared to densely populated 2D ribbons. Furthermore, five participants stated that the Instant Menu occupies far lesser space than the ribbons, and so they will not have to pan much even at high zoom levels.

Regarding the automatic focus-adjust feature of MagPro, two different opinions emerged during the qualitative analysis of feedback data. Five participants preferred the continuous auto-adjustment of MagPro that automatically moves the magnifier focus after each update to the keyboard cursor location. The remaining four participants however, desired the adjustment of viewport to happen only when the current keyboard focus was at edge of the current viewport. When probed, these participants explained that they preferred a steady viewport to the maximum extent possible, and also that they found the continuous viewport adjustment a little disturbing to their eyes.

6 DISCUSSION

The performance improvement with MagPro can be directly ascribed to its design principle of "bringing relevant elements closer", achieved by presenting the ribbon commands in a compact form as close as possible to the user's current magnifier context. However, our study also revealed certain limitations of MagPro. A few notable limitations and the future work for addressing them are discussed next.

Limitations. One of the limitations of MagPro is the dependency on Microsoft Primary Interop Assembly to track the user's current focus in the main edit area as well as execute application commands on user's behalf. While the Primary Interop Assembly is exclusive to Microsoft productivity tools, there are equivalent libraries and APIs for other productivity tools from other vendors (e.g., Google Apps Script API provides the same support as the Primary Interop Assembly). Even if there is no explicitly offered library for a productivity application, we can rely on the operating system's native accessibility API (e.g., UI Automation) for supporting MagPro functionalities (described later). Another obvious limitation of our work is the small sample size of the study. While our study established the generic potential of MagPro in assisting a wide range of low-vision conditions, customizations required for individual conditions (e.g., glaucoma, pigmentosa, optic atrophy, leber congenital amaurosis) are still unclear. This will require more user studies with individual low-vision groups. Brief descriptions of future work to address these limitations are discussed next.

Generalization of MagPro to arbitrary applications. As mentioned earlier, in the absence of supporting libraries like the Primary Interop Assembly, native accessibility APIs can be leveraged to provide the functionalities of MagPro. The accessibility APIs (e.g., UI Automation in Microsoft Windows platform) provide access to the entire UI tree of the applications, where each application element is represented as a tree node. By identifying nodes that correspond to the application commands, we can simulate clicks on these commands by simply calling *node.click()* or *node.expand()*, thereby serving as an effective replacement for Interop Assembly. However, accurately identifying the command nodes is a technical challenge, which is the scope of our future research.

Personalization for specific low-vision conditions. The current prototype of MagPro is still a 'one-size-fits-all' solution. However, low vision is heterogeneous in that it encompasses a wide array of low-vision conditions. For example, individuals who require lower zoom levels may prefer commands be arranged in multiple columns side-by-side in the MagPro's Instant Menu so as to reduce the number of scrolls needed to find the desired command. Therefore, customization or configuration options need to be provided for users to tailor MagPro to accommodate their individual needs and preferences. To understand the needs of individual low-vision groups, we plan to conduct usability studies specifically targeting the individual low-vision groups. The observations from these studies will then drive the optimization of MagPro for different low-vision conditions.

7 CONCLUSION

Interaction with productivity applications using screen magnifiers is tedious and stressful for people with low vision, as confirmed by the findings of our usability study with 10 participants having different low-vision conditions. To address this, we presented MagPro, an easy-to-use augmentation to the application that can significantly reduce the panning and zooming effort by enabling WYSIWYG (What You See Is What You Get) feedback while doing tasks in productivity tools. An evaluation of MagPro with nine participants showed significant improvement in the efficiency and usability while doing tasks, thereby demonstrating its potential in enhancing the user experience for a wide range of low-vision users.

ACKNOWLEDGMENTS

This work was supported by NSF Awards: 1805076, 1936027, NIH Awards: R01EY026621, R01EY030085, R01HD097188, NIDILRR Award: 90IF0117-01-00.

REFERENCES

[1] Aries Arditi and Jianwei Lu. 2008. Accessible Web Browser Interface Design for Users with Low Vision. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 52, 6 (2008), 576–580. https://doi.org/10.1177/154193120805200614 arXiv:https://doi.org/10.1177/154193120805200614

[2] Edward C Bell and Natalia M Mino. 2015. Employment outcomes for blind and visually impaired adults. (2015).

[3] Jeffrey P. Bigham. 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* (Honolulu, Hawaii, USA) *(UIST '14)*. ACM, New York, NY, USA, 117–122. https://doi.org/10.1145/2642918.2647357

[4] Syed Masum Billah, Vikas Ashok, Donald E. Porter, and I.V. Ramakrishnan. 2018. SteeringWheel: A Locality-Preserving Magnification Interface for Low Vision Web Browsing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 20, 13 pages. https://doi.org/10.1145/3173574.3173594

[5] Yevgen Borodin, Jeffrey P. Bigham, Glenn Dausch, and I. V. Ramakrishnan. 2010. More than Meets the Eye: A Survey of Screen-Reader Browsing Strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)* (Raleigh, North Carolina) *(W4A '10)*. Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages. https://doi.org/10.1145/1805986.1806005

[6] James V. Bradley. 1958. Complete Counterbalancing of Immediate Sequential Effects in a Latin Square Design. *J. Amer. Statist. Assoc.* 53, 282 (1958), 525–528. https://doi.org/10.1080/01621459.1958.10501456 arXiv:https://amstat.tandfonline.com/doi/pdf/10.1080/01621459.1958.10501456

[7] Julian Brinkley and Nasseh Tabrizi. 2017. A Desktop Usability Evaluation of the Facebook Mobile Interface using the JAWS Screen Reader with Blind Users. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 61, 1 (2017), 828–832. https://doi.org/10.1177/1541931213601699 arXiv:https://doi.org/10.1177/1541931213601699

[8] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[9] D. G. Evans, T. Diggle, S. H. Kurniawan, and P. Blenkhorn. 2003. An investigation into formatting and layout errors produced by blind word-processor users and an evaluation of prototype error prevention and correction techniques. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11, 3 (Sept 2003), 257–268. https://doi.org/10.1109/TNSRE.2003.816868

[10] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically generating personalized user interfaces with Supple. *Artificial Intelligence* 174, 12 (2010), 910 – 950. https://doi.org/10.1016/j.artint.2010.05.005

[11] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. 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* (Newport, Rhode Island, USA) *(UIST '07)*. ACM, New York, NY, USA, 231–240. https://doi.org/10.1145/1294211.1294253

[12] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139 – 183. https://doi.org/10.1016/S0166-4115(08)62386-9

[13] Apple Inc. 2020. Change Zoom preferences for accessibility on Mac - Apple Support. https://support.apple.com/guide/mac-help/change-zoom-preferences-for-accessibility-mh40579/mac.

[14] Julie A. Jacko, Armando B. Barreto, Gottlieb J. Marmet, Josey Y. M. Chu, Holly S. Bautsch, Ingrid U. Scott, and Robert H. Rosa, Jr. 2000. Low Vision: The Role of Visual Acuity in the Efficiency of Cursor Movement. In *Proceedings of the*

Fourth International ACM Conference on Assistive Technologies (Arlington, Virginia, USA) *(Assets '00)*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/354324.354327

[15] Richard L. Kline and Ephraim P. Glinert. 1995. Improving GUI Accessibility for People with Low Vision. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 114–121. https://doi.org/10.1145/223904.223919

[16] Thomas Knack. 2012. Real-time content-aware video retargeting on the Android platform for tunnel vision assistance. (2012).

[17] Sri Hastuti Kurniawan, Alistair G Sutcliffe, and Paul Blenkhorn. 2003. How Blind Users' Mental Models Affect Their Perceived Usability of an Unfamiliar Screen Reader. In *INTERACT*, Vol. 3. 631–638.

[18] Jonathan Lazar, Aaron Allen, Jason Kleinman, and Chris Malarkey. 2007. What frustrates screen reader users on the web: A study of 100 blind users. *International Journal of human-computer interaction* 22, 3 (2007), 247–269.

[19] Jingyi Li, Son Kim, Joshua A. Miele, Maneesh Agrawala, and Sean Follmer. 2019. Editing Spatial Layouts Through Tactile Templates for People with Visual Impairments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 206, 11 pages. https://doi.org/10.1145/3290605.3300436

[20] George Mamaladze. 2015. GlobalMouseKeyHook. https://github.com/gmamaladze/globalmousekeyhook.

[21] Natalie Maus, Dalton Rutledge, Sedeeq Al-Khazraji, Reynold Bailey, Cecilia Ovesdotter Alm, and Kristen Shinohara. 2020. Gaze-Guided Magnification for Individuals with Vision Impairments. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382995

[22] Nihal Menzi-Çetin, Ecenaz Alemdağ, Hakan Tüzün, and Merve Yıldız. 2017. Evaluation of a University Website's Usability for Visually Impaired Students. *Univers. Access Inf. Soc.* 16, 1 (March 2017), 151–160. https://doi.org/10.1007/s10209-015-0430-3

[23] Microsoft. 2017. Office solutions development overview (VSTO). https://docs.microsoft.com/en-us/visualstudio/vsto/office-solutions-development-overview-vsto?view=vs-2019.

[24] Microsoft. 2019. Office primary interop assemblies - Visual Studio. https://docs.microsoft.com/en-us/visualstudio/vsto/office-primary-interop-assemblies?view=vs-2019.

[25] Microsoft. 2020. Use Magnifier to make things on the screen easier to see - Windows Help. https://support.microsoft.com/en-us/help/11542/windows-use-magnifier-to-make-things-easier-to-see.

[26] Lourdes Morales, Sonia M. Arteaga, and Sri Kurniawan. 2013. Design Guidelines of a Tool to Help Blind Authors Independently Format Their Word Documents. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (Paris, France) *(CHI EA '13)*. ACM, New York, NY, USA, 31–36. https://doi.org/10.1145/2468356.2468363

[27] Lourdes Moreno, Xabier Valencia, J. Eduardo Pérez, and Myriam Arrue. 2018. Exploring the Web Navigation Strategies of People with Low Vision. In *Proceedings of the XIX International Conference on Human Computer Interaction* (Palma, Spain) *(Interacción 2018)*. Association for Computing Machinery, New York, NY, USA, Article 13, 8 pages. https://doi.org/10.1145/3233824.3233845

[28] Giulio Mori, Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Victor M. R. Penichet. 2011. Making "Google Docs" User Interface More Accessible for Blind People. In *Advances in New Technologies, Interactive Interfaces, and Communicability*. Springer Berlin Heidelberg, Berlin, Heidelberg, 20–29.

[29] Thomas Rathfux, Jasmin Thöner, Hermann Kaindl, and Roman Popp. 2018. Combining Design-Time Generation of Web-Pages with Responsive Design for Improving Low-Vision Accessibility. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Paris, France) *(EICS '18)*. Association for Computing Machinery, New York, NY, USA, Article 10, 7 pages. https://doi.org/10.1145/3220134.3220141

[30] Johnny Saldaña. 2015. *The coding manual for qualitative researchers*. Sage.

[31] Andreas Savakis, Mark Stump, Grigorios Tsagkatakis, Roy Melton, Gary Behm, and Gwen Sterns. 2012. Low vision assistance using face detection and tracking on android smartphones. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 1176–1179.

[32] Freedom Scientific. 2020. ZoomText Screen Magnifier and Screen Reader - zoomtext.com. https://www.zoomtext.com/.

[33] Lee Stearns, Victor DeSouza, Jessica Yin, Leah Findlater, and Jon E. Froehlich. 2017. Augmented Reality Magnification for Low Vision Users with the Microsoft Hololens and a Finger-Worn Camera. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, Maryland, USA) *(ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 361–362. https://doi.org/10.1145/3132525.3134812

[34] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. 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* (Reno, Nevada, USA) *(ASSETS '16)*. ACM, New York, NY, USA, 171–180. https://doi.org/10.1145/2982142.2982168

[35] Katie Wang, Laura G Barron, and Michelle R Hebl. 2010. Making those who cannot see look best: Effects of visual resume formatting on ratings of job applicants with blindness. *Rehabilitation psychology* 55, 1 (2010), 68.

[36] Hui-Yin Wu, Aurelie Calabrese, and Pierre Kornprobst. 2020. *Towards Accessible News Reading Design in Virtual Reality for Low Vision.* Research Report RR-9298. UCA ; Inria. 20 pages. https://hal.inria.fr/hal-02321739

[37] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300341

[38] Yuhang Zhao, Elizabeth Kupferstein, Doron Tal, and Shiri Azenkot. 2018. "It Looks Beautiful but Scary": How Low Vision People Navigate Stairs and Other Surface Level Changes. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (Galway, Ireland) *(ASSETS '18)*. Association for Computing Machinery, New York, NY, USA, 307–320. https://doi.org/10.1145/3234695.3236359

Received April 2020; revised July 2020; accepted October 2020