

2020

## Rotate-and-Press: A Non-Visual Alternative to Point-and-Click

Hae-Na Lee

Vikas Ashok  
*Old Dominion University*

I. V. Ramakrishnan

Follow this and additional works at: [https://digitalcommons.odu.edu/computerscience\\_fac\\_pubs](https://digitalcommons.odu.edu/computerscience_fac_pubs)



Part of the [Disability Studies Commons](#), [Graphics and Human Computer Interfaces Commons](#), and the [Sense Organs Commons](#)

---

### Original Publication Citation

Lee, H. N., Ashok, V., & Ramakrishnan, I. V. (2020, Jul) Rotate-and-press: A non-visual alternative to point-and-click? In: Stephanidis C., Antona M., Gao Q., Zhou J. (eds) HCI International 2020 – Late Breaking Papers: Universal Access and Inclusive Design. HCII 2020 July 19-24, 2020, Copenhagen, Denmark. Lecture Notes in Computer Science, vol 12426. Springer, Cham. [https://doi.org/10.1007/978-3-030-60149-2\\_23](https://doi.org/10.1007/978-3-030-60149-2_23)

This Conference Paper is brought to you for free and open access by the Computer Science at ODU Digital Commons. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of ODU Digital Commons. For more information, please contact [digitalcommons@odu.edu](mailto:digitalcommons@odu.edu).



# HHS Public Access

Author manuscript

*HCI Int 2020 Late Break Posters (2020)*. Author manuscript; available in PMC 2021 March 02.

Published in final edited form as:

*HCI Int 2020 Late Break Posters (2020)*. 2020 July ; 12426: 291–305.

doi:10.1007/978-3-030-60149-2\_23.

## Rotate-and-Press: A Non-visual Alternative to Point-and-Click?

Hae-Na Lee<sup>1</sup>, Vikas Ashok<sup>2</sup>, I. V. Ramakrishnan<sup>1</sup>

<sup>1</sup>Stony Brook University, Stony Brook, NY 11794, USA

<sup>2</sup>Old Dominion University, Norfolk, VA 23529, USA

### Abstract

Most computer applications manifest visually rich and dense graphical user interfaces (GUIs) that are primarily tailored for an easy-and-efficient sighted interaction using a combination of two default input modalities, namely the keyboard and the mouse/touchpad. However, blind screen-reader users predominantly rely only on keyboard, and therefore struggle to interact with these applications, since it is both arduous and tedious to perform the visual ‘point-and-click’ tasks such as accessing the various application commands/features using just keyboard shortcuts supported by screen readers.

In this paper, we investigate the suitability of a ‘rotate-and-press’ input modality as an effective non-visual substitute for the visual mouse to easily interact with computer applications, with specific focus on word processing applications serving as the representative case study. In this regard, we designed and developed bTunes, an add-on for Microsoft Word that customizes an off-the-shelf Dial input device such that it serves as a *surrogate* mouse for blind screen-reader users to quickly access various application commands and features using a set of simple rotate and press gestures supported by the Dial. Therefore, with bTunes, blind users too can now enjoy the benefits of two input modalities, as their sighted counterparts. A user study with 15 blind participants revealed that bTunes significantly reduced both the time and number of user actions for doing representative tasks in a word processing application, by as much as 65.1% and 36.09% respectively. The participants also stated that they did not face any issues switching between keyboard and Dial, and furthermore gave a high usability rating (84.66 avg. SUS score) for bTunes.

### Keywords

Screen reader; Word processor; Accessibility

## 1 Introduction

People who are blind generally rely on special-purpose assistive technology, namely screen readers (e.g., JAWS [10], VoiceOver [2], NVDA [18]), for interacting with computing applications. A screen reader linearly narrates contents of the screen, and also enables blind users to navigate the application GUI using predefined keyboard hotkeys or shortcuts. The

primary input device for blind users to interact with computer applications using screen readers, is a keyboard. However, most applications manifest visually dense GUIs that are more suited for interaction with a visual pointing device such as a mouse or touchpad. For example, in Microsoft Word, as shown in Fig. 1, to apply a command (e.g., Styles) while editing a document, sighted users can simply move the mouse cursor to that command in the ribbon and click on it. On the contrary, to do the same task, blind users have to either memorize the corresponding shortcut or serially move their screen-reader focus to the command by pressing a multitude of basic navigational keyboard shortcuts. Therefore, tasks that the sighted users can perform almost instantaneously with a simple point-and-click mouse operation, are tedious and cumbersome for blind users using just the keyboard.

Prior approaches [1,5,6,14,25] devised to mitigate this usability divide have primarily focused on passive content navigation or ‘consumption’, especially in web browsing, accessing maps and graph charts. However, interaction with most general computer applications, especially productivity tools, goes much beyond just content navigation; users also need to frequently access various application commands and features (e.g., formatting, insertions, review, comments, etc., in Microsoft Word) while they navigate or edit the main content. To fill this gap, in this paper, we investigate the suitability and potential of a ‘rotate-and-press’ input modality as an effective non-visual substitute or ‘surrogate’ for the visual mouse to enable blind screen-reader users to easily and efficiently access application commands and features while they interact with the main content.

With the additional tangible rotary input modality, blind screen-reader users too will be able to benefit from having two input modalities akin to their sighted peers. For instance, in productivity applications, sighted users can effectively distribute their interaction load over both keyboard and mouse, e.g., using the keyboard for typing and pressing some hotkeys, and using the mouse for instantly accessing application commands. Blind users on the other hand, have to rely solely on the keyboard to do all the tasks. Given the linear content-navigation supported by screen readers, blind users find it tedious and cumbersome to perform even simple tasks such as accessing application commands. However, with the auxiliary ‘rotate-and-press’ input device, blind users too will be able to effectively split their workload over two input modalities and complete their tasks quickly and easily.

As an investigation tool, we developed bTunes. We chose Microsoft Word as a use scenario due to its popularity among blind users [23] and also its sophisticated GUI containing a variety of application commands. bTunes adapts an off-the-shelf rotary input device, namely Microsoft Surface Dial (see Fig. 1) to serve as a “surrogate mouse”, thereby providing an auxiliary tangible interface in addition to keyboard for blind users. As shown in Fig. 1, via simple rotate and press gestures supported by the Dial, bTunes enables a user to easily access all the ribbon commands, without losing their current keyboard context in the main content area of the document. Results from a user study with 15 blind participants were very encouraging in that the time and number of user actions the participants needed for accessing commands with bTunes were significantly reduced by as much as 65.1% and 36.09%, respectively, when compared to their current status quo.

We summarize our contributions as follows:

- The design and implementation of bTunes – an add-on for word processing applications, specifically Microsoft Word, which enables blind users to easily and efficiently access application commands and features using a ‘rotate-and-press’ interaction modality, thereby enhancing the productivity of blind users with these applications.
- Results from a user study with 15 blind screen-reader users that demonstrated the potency of bTunes in significantly improving the user experience with word processing applications.

## 2 Related Work

To overcome the limitations of keyboard-based screen-reader interaction, several non-visual input modalities for blind users have been previously explored [1,3,5,6,14,19–21,24,25]. Broadly, these approaches can be grouped into keyboard adaptation [3,15], audio-tactile devices [6,14,21,25], and assistant interfaces [4,5,11,16].

Keyboard adaptation approaches repurpose the keyboard to improve interaction experience for blind users. In the IBM Home Page Reader (HPR) [3], the numeric keypad was adapted to serve as an auxiliary input interface for navigating web pages. Khurana et al. [15], on the other hand, propose spatially region interaction techniques that leverage the keyboard surface to facilitate easily non-visual interaction with 2D structures. Besides the need to remember new shortcuts on top of the multiple existing screen-reader shortcuts, both these approaches are exclusive to web browsing, and therefore do not readily generalize to arbitrary computer applications, such as Word supported by bTunes.

Audio-haptic approaches enable screen-reader users to leverage additional tangible audio-tactile input devices to interact with applications. For example, the multimodal audio-haptic interface proposed by Doush et al. [1] enables screen-reader users to navigate and access content in Excel charts. Perhaps the closest related work is the Speed-Dial [6], which supports easy hierarchical navigation of webpage content via its external Microsoft Surface Dial input interface. Also, Soviak et al. [21] present an audio-haptic glove that helps blind users to feel the borders of various webpage segments on the page, thereby giving the users a sense of page layout and content arrangement. A common aspect of all these approaches is that they are designed exclusively for passive content navigation, which is different from interaction with general applications such as Word, where the users not only navigate content, but also frequently accessing the various spatially-distributed application commands and features (e.g., formatting, insertions, review, comments, etc.).

Assistants let blind users interact with applications using spoken commands. For example, the assistant proposed by Gadde et al. [11] lets blind users to rely on a few speech commands to get a quick overview of current webpage and also to navigate to a section of interest. On the other hand, Ashok et al. [5] support a richer set of voice commands that lets blind users also query the webpage content. While speech assistants are known to significantly improve usability for blind users, they have to be custom-designed for each application. The general-purpose assistants like Apple’s Siri, Microsoft’s Cortana, etc. primarily focus on OS-level commands (e.g., open an application, simulate mouse and

keyboard actions, open windows menu, set up alarms, etc.), factoid queries (e.g., time, weather, etc.), and dictation (e.g., insert paragraph, edit word, delete line, etc.). They are presently incapable of providing speech access to the various commands supported within arbitrary applications. Lastly, speech assistants including commercial ones only support a limited set of languages.

Proficiency with word processing applications has been recognized as an important skill for employment of blind individuals [8,22]. Despite the importance of these applications and in contrast to the large body of work on the accessibility of the Web and mobile devices as noted above, there is a dearth of studies on usability of desktop applications, in particular the Office suite [1,17]. Furthermore, none of them focus on understanding user behavior and interaction strategies that blind people employ to create and edit documents. Apple's MacBook Pro Touch Bar [24] is a generic solution that provides contextual menus and navigation shortcuts for arbitrary computer applications. However, the Touch Bar can only contain a few commands, and moreover it is primarily designed for visual consumption, thereby requiring screen-reader users to spend significant time exploring and orienting themselves each time they want to access the features on it. Like Touch Bar, Apple's built-in screen reader, VoiceOver, also provides access to commands via its rotor feature. However, these commands mainly assist in navigating content.

Perhaps the closest work related to this paper is [17], where the authors suggest guidelines for a support tool in Microsoft Word that can assist blind people format their documents independently. However, these guidelines were developed based only on subjective feedback obtained from a preliminary survey with 15 blind users, and therefore did not incorporate objective details regarding user-interaction behavior and strategies. Evans et al. [9] also proposed a technique to assist blind users format documents properly in Word. They first checked the post-interaction documents produced by blind users to figure out common layout and formatting errors, and then based on their observations, built two prototypes to help the blind users detect and rectify errors.

### 3 bTunes Design

Figure 2 presents an architectural overview of bTunes designed for Microsoft Word application. As shown in the figure, with bTunes, blind screen-reader users have an additional input modality, namely Dial, to access various application commands anytime without having to manually move the keyboard focus away from their current context in the main work-area of the application. These commands correspond to the non-edit word-processing actions such as formatting, commenting, proof-reading, inserting objects, changing design, and so on. bTunes replicates the command structure of Word (i.e., ribbons) in the Dial's radial menu (see Fig. 1) and establishes one-to-one programmatic hooks between the commands in the bTunes interface and the corresponding commands in the application GUI. This way, selecting a command with bTunes emulates selecting the corresponding one in the application GUI, thereby producing the same intended outcome. For commands with options (e.g., font names for the Font command), bTunes refreshes its dialog box to show these options in place of commands (see Fig. 1). The users can access,

navigate, and select ribbons and commands in the radial menu and the dialog box using simple rotate and press gestures, as explained later in this section.

### 3.1 Dial Input Device

The off-the-shelf Surface Dial input device (shown in Fig. 1) is a small rotary puck that supports three simple gestures: *press*, *rotate*, and *press-and-hold*. We also implemented a *double press* gesture, which is triggered when the Dial is pressed twice in quick succession (less than 400 ms). On every gesture, the Dial also provides tactile feedback in the form of vibrations. The Surface Dial is usable with a PC running Windows 10 Anniversary Update or later, and it gets connected to the PC via Bluetooth 4.0 LE.

### 3.2 bTunes Interaction Using Gestures

A simple press-and-hold gesture brings up the radial dashboard containing outer command ribbons (i.e., Home, Insert, Design, etc.). A user can perform rotate gestures to access the desired command ribbon, and then execute a single press to shift focus to the inner ribbon containing commands, which is shown in a separate dialog box. In this dialog box, the user can do rotate gestures to access different commands, followed by a press gesture to execute the desired command. If the command has options, the press gesture will refresh the dialog box with the corresponding list of options, and the user can repeat the process of using the rotate gestures to navigate to desired option and then the press gesture to select the desired option (e.g., Font Size). At any instant, a double press gesture shifts focus back one level, i.e., from options list to inner ribbon commands, or from inner ribbon to outer ribbon group. A double press at outer ribbon will automatically close the bTunes interface and the focus will shift back to the main work area. The user can also press a shortcut or simply type at anytime to instantly close the bTunes interface.

### 3.3 Implementation Details

We implemented bTunes as a Microsoft Word add-in, by utilizing the services of the Office Word Primary Interop Assembly (PIA)<sup>1</sup>. Specifically, we developed bTunes with the Visual C# under Visual Studio .NET Framework 4.6.1. We utilized Visual Studio Tools for Office (VSTO) Add-in<sup>2</sup> to build custom Dial operations and radial menu for the bTunes components.

The current bTunes prototype can be easily adapted for any Office productivity application using the corresponding PIA. For arbitrary applications, bTunes can be adapted to leverage the UI Automation accessibility framework [13] instead of Interop services, to obtain the UI composition of any application in the form of a tree, and then automatically identify and enable users to easily and hierarchically navigate the application command ‘tree’ using Dial gestures.

---

<sup>1</sup><https://docs.microsoft.com/en-us/visualstudio/vsto/office-primary-interopassemblies?view=vs-2019>.

<sup>2</sup><https://docs.microsoft.com/en-us/visualstudio/vsto/office-solutions-development-overview-vsto?view=vs-2017>.

## 4 Evaluation

### 4.1 Participants

For the study, we recruited 15 fully blind participants (6 female, 9 male) through local mailing lists and word-of-mouth. The participants varied in age between 31 and 63 (Mean = 47.86, Median = 46, SD = 11.06). All participants stated that they were either blind by birth or lost eyesight at a very young age (less than 10 years old). None of the participants had any motor impairments that affected their physical interaction with the Dial input device. The inclusion criteria required the participants to be proficient with Microsoft Word and JAWS screen reader. All participants stated that they frequently used Office productivity applications, file explorer, web browsers, communication software, and control panel settings. A few participants also frequently used Integrated Development Environments (IDEs), statistical tools, media players, and music software. Table 1 presents the participant demographics.

### 4.2 Apparatus

The study was performed using ASUS ROG GU501 laptop with Windows 10, Microsoft Word, and JAWS screen reader installed. Also, an external standard keyboard and Microsoft Surface Dial were connected to the laptop.

### 4.3 Design

The study required the participants to do the following two tasks:

- **Task 1:** Find and apply a command in the Microsoft Word application.
- **Task 2:** Create an article with a title, a heading, and two paragraphs.

The participants were asked to perform these representative tasks under the following two study conditions:

- **Screen Reader:** Participants used only the JAWS keyboard shortcuts to do the tasks.
- **bTunes:** Participants used both the JAWS keyboard shortcuts and the bTunes's Dial interface (e.g., press, rotate, and double press gestures) to do the tasks.

Task 1 was controlled as it was designed to compare the command-access efficiencies of screen reader and bTunes, whereas Task 2 was think-aloud free-form editing as it was intended to measure perceived overall usability of screen reader and bTunes in a reasonably realistic setting. For Task 1, we chose the following six commands: (a) Set *Text Highlight Color* to 'Dark Blue' in Home ribbon; (b) Insert *Star: 5 Points* shape in Insert ribbon; (c) Set *Page Color* to 'Light Blue' in Design ribbon; (d) Set *Position Object* to 'Bottom Right' in Layout ribbon; (e) Select *Bibliography Style* to 'MLA' in Bibliography ribbon; and (f) Configure *Markup options* to 'Show All Revisions Inline' in Review ribbon. For Task 2, we chose the following two topics: (a) school; and (b) their neighborhood.

In each condition, the participants accessed three commands for Task 1, and created one article for Task 2. To minimize learning effects, the assignment of commands and articles to

tasks was randomized, and the ordering of tasks and conditions were counterbalanced. Also, to avoid confounds, for Task 1, we selected commands that are equidistant from the beginning of their corresponding ribbons (i.e., 23<sup>rd</sup> command considering the linear screen-reading navigation order in each ribbon), and hence would require the same number of basic <Tab> shortcuts or rotate gestures to navigate to them.

#### 4.4 Procedure

The experimenter began the study by demonstrating the bTunes's Dial interface to the participants and letting them practice for 10 min to get comfortable with bTunes. The experimenter then let the participants practice with JAWS screen reader for 10 min and refresh their memory about the various available shortcuts. After the practice session, the participant performed the tasks according to a predetermined counterbalanced order. Post study, the experimenter surveyed the participant with the System Usability Scale (SUS), NASA Task Load Index (NASA-TLX), and custom questionnaires. Each study lasted for 1–1.5 h, and all conversations were in English.

**Measurements.**—During the study, the experimenter measured task completion times, and logged all screen-reader keystrokes and Dial gestures. Audio and computer-screen activities were recorded using the *Open Broadcaster Software*. The experimenter also took notes while the participants were doing the tasks. At the end of the study, the experimenter administered the System Usability Scale (SUS), NASA Task Load Index (NASA-TLX), and a custom open-ended questionnaire to collect subjective feedback.

#### 4.5 Results

**Completion Times and User Effort for Task 1.**—Figure 3 presents the task completion times and number of user actions for Task 1 under both conditions. As shown in the figure, overall, the participants spent an average of 171.44 s (Median = 159, Max = 600, Min = 10) with screen reader, whereas they only needed an average of 59.97 s (Median = 53, Max = 144, Min = 25) with bTunes. A Wilcoxon signed rank test showed a significant difference in the command-access times between the two study conditions ( $z = -5.197$ ,  $n = 45$ ,  $p < 0.00001$ ). Similar observations were made while analyzing the completion times for individual participant groups, i.e., beginner (Mean = 242.76, Median = 216, Max = 600, Min = 32) and expert (Mean = 109.04, Median = 81.5, Max = 287, Min = 10). We found significant effect of study conditions on completion times for both beginner ( $W = 1 < 58$ ,  $n = 21$ ) and expert ( $W = 47 < 84$ ,  $n = 24$ ) groups. However, between the two groups, the experts were significantly faster than beginners in accessing commands with screen reader (Mann Whitney U test,  $U_1 = 88.5$ ,  $U_2 = 415.5$ ,  $p = 0.0001$ ), but no such significant difference was found while accessing commands with bTunes ( $U_1 = 289.5$ ,  $U_2 = 214.5$ ,  $p = 0.393$ ).

Also, with screen reader, overall, the participants spent an average of 48.57 shortcuts (Median = 51, Max = 70, Min = 25), whereas with bTunes, they only used an average of 31.04 gestures (Median = 32, Max = 35, Min = 25). This difference in input effort was found to be statistically significant (Wilcoxon signed rank test,  $|z| = 5.48 < z_c(1.96)$ ,  $n = 45$ ). We also found significant effect of study conditions on number of shortcuts/gestures for both beginner ( $W = 0 < 58$ ,  $n = 21$ ) and expert ( $W = 0 < 81$ ,  $n = 24$ ) groups. As in case of task

completion times, with screen reader, the experts needed significantly fewer shortcuts (Mann Whitney U test,  $U_1 = 36.5$ ,  $U_2 = 467.5$ ,  $p < 0.0001$ ) than beginners to access commands; however, no such significant difference was observed with bTunes ( $U_1 = 246$ ,  $U_2 = 258$ ,  $p = 0.89$ ).

We did not measure the task completion times for Task 2, as it involved uncontrolled think-aloud free-form editing, thereby making the task completion times incomparable between conditions.

**Subjective Feedback.**—At the end of each study session, every participant was administered the standard System Usability Scale (SUS) questionnaire [7] 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 being neutral. Overall, we found a significant difference in the SUS scores between bTunes ( $\mu = 84.66$ ,  $\sigma = 5.07$ ) and screen reader ( $\mu = 57.5$ ,  $\sigma = 17.46$ ) conditions (paired t-test,  $|t| = 6.741 > 2.145$ ,  $df = 14$ ). The difference in average SUS scores was also statistically significant within both beginner (screen reader:  $\mu = 46.07$ ,  $\sigma = 12.94$ , bTunes:  $\mu = 82.5$ ,  $\sigma = 4.62$ ), and expert (screen reader:  $\mu = 67.5$ ,  $\sigma = 14.52$ , bTunes:  $\mu = 86.56$ ,  $\sigma = 4.66$ ) groups ( $|t| = 7.47 > 2.447$ ,  $df = 6$  for beginners, and  $|t| = 3.977 > 2.365$ ,  $df = 7$  for experts). However, between experts and beginners, the experts rated the screen reader significantly higher than beginners (t-test unequal variances,  $|t| = 3.021 > 2.161$ ,  $df = 12.98$ ,  $p = 0.0098$ ); however, no such difference in ratings was observed for bTunes ( $|t| = 1.692 < 2.164$ ,  $df = 12.76$ ,  $p = 0.1149$ ).

We also administered the widely used NASA-TLX [12] subjective questionnaire for assessing perceived task workload (expressed as a value between 0 and 100, with lower values indicating better results). Overall, we found a significant difference in the TLX scores between screen reader ( $\mu = 59.97$ ,  $\sigma = 14.11$ ) and bTunes ( $\mu = 17.35$ ,  $\sigma = 2.55$ ) conditions (paired t-test,  $|t| = 11.92 > 2.145$ ,  $df = 14$ ). The difference in average TLX scores was also statistically significant within both beginner (screen reader:  $\mu = 73.95$ ,  $\sigma = 3.15$ , bTunes:  $\mu = 18.42$ ,  $\sigma = 2.72$ ), and expert (screen reader:  $\mu = 47.75$ ,  $\sigma = 6.66$ , bTunes:  $\mu = 16.41$ ,  $\sigma = 1.96$ ) participant groups (t-test,  $|t| = 28.71 > 2.447$ ,  $df = 6$  for beginners, and  $|t| = 12.7 > 2.365$ ,  $df = 7$  for experts). However, between experts and beginners, the perceived workload of beginners with screen readers was significantly higher than that of experts (t-test unequal variances,  $|t| = 9.931 > 2.221$ ,  $df = 10.255$ ,  $p < 0.001$ ), however, no such difference was observed in case of bTunes ( $|t| = 1.622 < 2.206$ ,  $df = 10.794$ ,  $p = 0.133$ ).

**Qualitative Feedback for Task 2.**—All participants indicated that they did not have any problems switching between the keyboard and the Dial in bTunes while doing the tasks. On the contrary, they stated they preferred this clear separation of interaction activities, i.e., using the keyboard for typing and pressing few hotkeys, and using the Dial for the accessing the application command and features. They also agreed that bTunes gestures were much simpler, natural, and easier to memorize compared to the screen-reader keyboard shortcuts. Eight participants (P2, P5, P6, P8, P9, P10, P13, and P15) stated that they frequently mix-up the screen-reader's shortcuts for different applications, and therefore waste valuable time due to these recurrent mistakes. However, they indicated that they would never run into such an issue with bTunes, as they don't have to rely on keyboard for doing actions.

Five participants (P2, P6, P9, P12, and P13) also stated that they preferred the small size of Dial input device compared to the large size of keyboard. These participants expressed that they especially liked the Dial interface because it allowed them to easily perform input actions with one hand, in contrast to keyboard where they often have to rely on both their hands to execute complex hotkeys (e.g., ALT + NUMPAD 5 in JAWS). They also indicated that with keyboard, there was a good chance of unintentionally pressing the wrong hotkeys especially when the keyboard buttons involved were far apart from each other; such problems will not occur with the Dial interface of bTunes.

Twelve participants (except P1, P4, and P7) noted that the bTunes interface is ‘smooth’ and straightforward when accessing the ribbon commands. In contrast, they stated that ribbon access is confusing with keyboard as there are multiple ways in which one can navigate the ribbon using a wide array of hotkeys. They also specified that with keyboard, it is easy to miss certain commands that cannot be accessed through generic shortcuts. For example, while doing Task 1, four participants (P2, P5, P10, and P13) navigated through the ribbon using the LEFT/RIGHT arrow keys, and therefore missed several commands that were only accessible by pressing TAB shortcut. Similarly, while accessing a grid of commands such as *Text Highlight Color*, 5 participants (P2, P5, P6, P10, and P14) initially pressed only the UP/DOWN arrow keys several times before realizing that they could access other colors by pressing the LEFT/RIGHT arrow keys. Furthermore, accidental key presses moved the screen-reader focus away from the ribbon, and therefore the participants had to repeat the tedious process of sequentially navigating a ribbon to find the task command. No such issues were observed with the Dial interface during the study.

## 5 Discussion

Our results clearly demonstrate the potential of bTunes in serving as an effective non-visual surrogate for visual pointing devices such as mouse and touchpad. The participants also gave higher usability rating for bTunes compared to their preferred keyboard-only screen reader. However, the study also revealed limitations and important avenues for future research, and we discuss a couple of the important ones next.

### Command Prediction.

Analysis of the study data revealed that further improvements in command access times and user effort can be achieved by predicting the commands that the user will most likely access next given their current application context, and then accordingly reordering the command list in the radial menu and the bTunes’s dialog box dynamically such that the most probable commands are placed at the beginning of this list. For example, in Word, commands such as *Alignment*, *Styles*, and *Font* are more likely to be applied on entire paragraphs or collections of paragraphs, compared to commands such as *Bold*, *Italic*, and *Underline* that are more likely to be used on small portions of text within a paragraph. Therefore, if the user highlights a paragraph, dynamically placing the former commands before the latter commands in the dialog box can potentially reduce the time and number of actions to access the desired command.

### Content Navigation.

While we focused only on accessing application commands and features in this paper, the rotate-and-press interaction modality can also be leveraged to support content navigation. For example, in Word, hierarchical navigation of content tree (i.e., section, subsection, and so on) can easily be supported using the rotate-and-press gestures; rotate to navigate nodes at the same level, single press to one level down the tree, and double press to go one level up. In 2D spreadsheets such as Microsoft Excel, the Dial interface can be used to go through the rows one-by-one using rotate gestures, and the Dial's radial menu can be used to access content in individual columns (e.g., age, date of birth, address, etc.). However, contrary to command access, content navigation requires semantic knowledge of the content layout and arrangement in order to provide an effective navigational interface. Automatically glean the semantics is a topic of future research.

### Generalizability of Implementation.

bTunes implementation can also be easily adapted for other Office productivity tools notably Excel, PowerPoint, Google Sheets, and Google Slides, as these tools too support interoperability services to access their metadata. For general desktop applications beyond office productivity tools, bTunes can leverage OS accessibility APIs (e.g., the UI Automation accessibility framework [13] for Windows) to obtain the UI composition of any application in the form of a tree, and then enable users to easily and hierarchically navigate this application 'tree' using Dial gestures. However, automatically glean the application semantics, and then accordingly customizing the bTunes interface for optimal user interaction, is a topic of future research.

## 6 Conclusion

This paper introduces a non-visual alternative to pointing devices, namely a 'rotate-and-press' Dial interface, to enhance blind users' interaction experience with computers. The paper also provides experimental evidence of the potential of bTunes in improving user satisfaction and experience while interacting with productivity applications, specifically word processors. It is anticipated that further research on this novel interaction paradigm will usher similar productivity and usability gains for all computing applications.

### Acknowledgments.

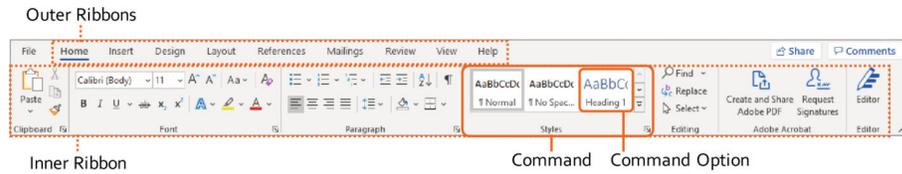
The work in this paper was supported by NSF Awards: 1805076, 1936027; NIH Awards: R01EY026621, R01EY030085, R01HD097188; and NIDILRR Award: 90IF0117-01-00.

### References

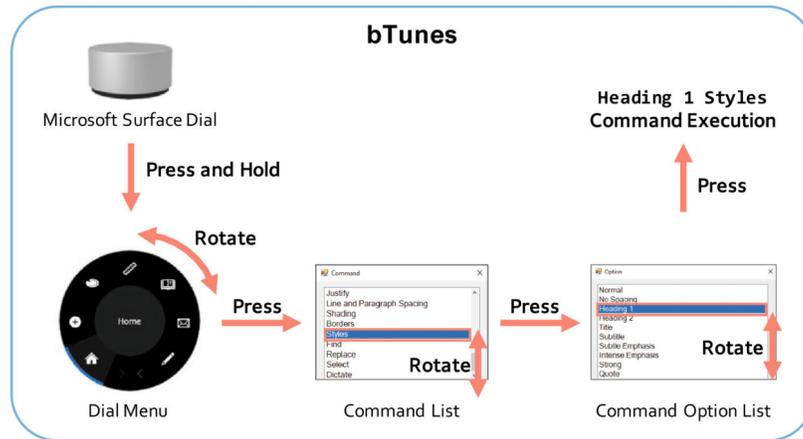
1. Abu Doush I, Pontelli E, Simon D, Son TC, Ma O: Making microsoft excel<sup>TM</sup>: multimodal presentation of charts. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, Assets 2009, New York, NY, USA, pp. 147–154. Association for Computing Machinery (2009). 10.1145/1639642.1639669
2. Apple Inc.: Vision accessibility - mac - apple (2020). <https://www.apple.com/accessibility/mac/vision/>

3. Asakawa C, Itoh T: User interface of a home page reader. In: Proceedings of the Third International ACM Conference on Assistive Technologies, Assets 1998, New York, NY, USA, pp. 149–156. ACM (1998). 10.1145/274497.274526
4. Ashok V, Borodin Y, Puzis Y, Ramakrishnan IV: Capti-speak: a speech-enabled web screen reader. In: Proceedings of the 12th Web for All Conference, W4A 2015, New York, NY, USA. Association for Computing Machinery (2015). 10.1145/2745555.2746660
5. Ashok V, Puzis Y, Borodin Y, Ramakrishnan I: Web screen reading automation assistance using semantic abstraction. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces, IUI 2017, New York, NY, USA, pp. 407–418. Association for Computing Machinery (2017). 10.1145/3025171.3025229
6. Billah SM, Ashok V, Porter DE, Ramakrishnan I: Speed-dial: a surrogate mouse for non-visual web browsing. In: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2017, New York, NY, USA, pp. 110–119. Association for Computing Machinery (2017). 10.1145/3132525.3132531
7. Brooke J, et al.: SUS-a quick and dirty usability scale. *Usab. Eval. Ind* 189(194), 4–7 (1996)
8. C. Bell E, M. Mino N: Employment outcomes for blind and visually impaired adults. *J. Blind. Innov. Res* 5 (2015). 10.5241/5-85
9. Evans DG, Diggie T, Kurniawan SH, Blenkhorn P: An investigation into formatting and layout errors produced by blind word-processor users and an evaluation of prototype error prevention and correction techniques. *IEEE Trans. Neural Syst. Rehabil. Eng* 11(3), 257–268 (2003). 10.1109/TNSRE.2003.816868 [PubMed: 14518789]
10. Freedom Scientific: Jaws ® - freedom scientific (2020). <http://www.freedomscientific.com/products/software/jaws/>
11. Gadde P, Bolchini D: From screen reading to aural glancing: towards instant access to key page sections. In: Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility, ASSETS 2014, New York, NY, USA, pp. 67–74. Association for Computing Machinery (2014). 10.1145/2661334.2661363
12. Hart SG, Staveland LE: Development of NASA-TLX (task load index): results of empirical and theoretical research. In: *Advances in Psychology*, vol. 52, pp. 139–183. Elsevier (1988)
13. Haverty R: New accessibility model for microsoft windows and cross platform development. *SIGACCESS Access. Comput* 82, 11–17 (2005). 10.1145/1077238.1077240
14. Jaijongrak V.r., Kumazawa I, Thiemjarus S: A haptic and auditory assistive user interface: helping the blinds on their computer operations. In: 2011 IEEE International Conference on Rehabilitation Robotics, pp. 1–6. IEEE (2011)
15. Khurana R, McIsaac D, Lockerman E, Mankoff J: Nonvisual interaction techniques at the keyboard surface. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, New York, NY, USA. Association for Computing Machinery (2018). 10.1145/3173574.3173585
16. Melnyk V, et al.: Look ma, no aria: generic accessible interfaces for web widgets. In: Proceedings of the 12th Web for All Conference, W4A 2015, New York, NY, USA. Association for Computing Machinery (2015). 10.1145/2745555.2746666
17. Morales L, Arteaga SM, Kurniawan S: Design guidelines of a tool to help blind authors independently format their word documents. In: CHI 2013 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2013, New York, NY, USA, pp. 31–36. ACM (2013). 10.1145/2468356.2468363
18. NV Access: Nv access (2020). <https://www.nvaccess.org/>
19. Owen JM, Petro JA, D’Souza SM, Rastogi R, Pawluk DT: An improved, low-cost tactile “mouse” for use by individuals who are blind and visually impaired. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, Assets 2009, New York, NY, USA, pp. 223–224. Association for Computing Machinery (2009). 10.1145/1639642.1639686
20. Rastogi R, Pawluk DT, Ketchum JM: Issues of using tactile mice by individuals who are blind and visually impaired. *IEEE Trans. Neural Syst. Rehabil. Eng* 18(3), 311–318 (2010) [PubMed: 20363685]

21. Soviak A, Borodin A, Ashok V, Borodin Y, Puzis Y, Ramakrishnan I: Tactile accessibility: does anyone need a haptic glove? In: Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2016, New York, NY, USA, pp. 101–109. Association for Computing Machinery (2016). 10.1145/2982142.2982175
22. Wang K, Barron LG, Hebl MR: Making those who cannot see look best: effects of visual resume formatting on ratings of job applicants with blindness. *Rehabil. Psychol* 55(1), 68 (2010) [PubMed: 20175636]
23. WebAIM: Screen reader user survey #8 results - webaim (2020). <https://webaim.org/projects/screenreadersurvey8/>
24. Wi cek-Janka E, Papierz M, Kornecka M, Nitka M: Apple products: a discussion of the product life cycle. In: 4th International Conference on Management Science and Management Innovation, vol. 31, pp. 159–164 (2017)
25. Zeng L, Miao M, Weber G: Interactive audio-haptic map explorer on a tactile display. *Interact. Comput* 27(4), 413–429 (2015)



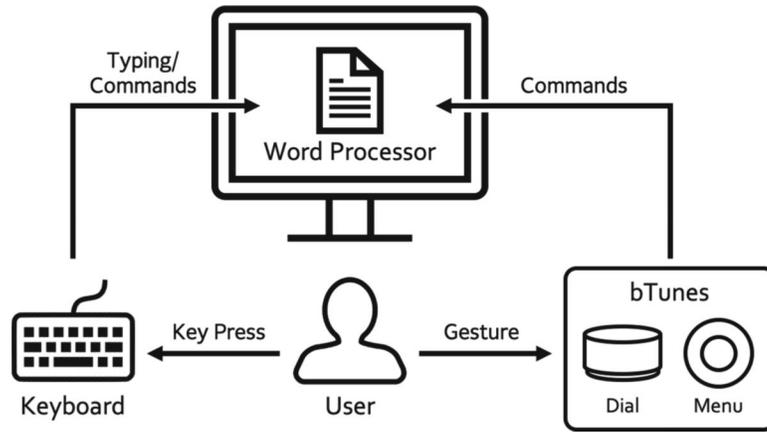
(a) Ribbons in Microsoft Word.



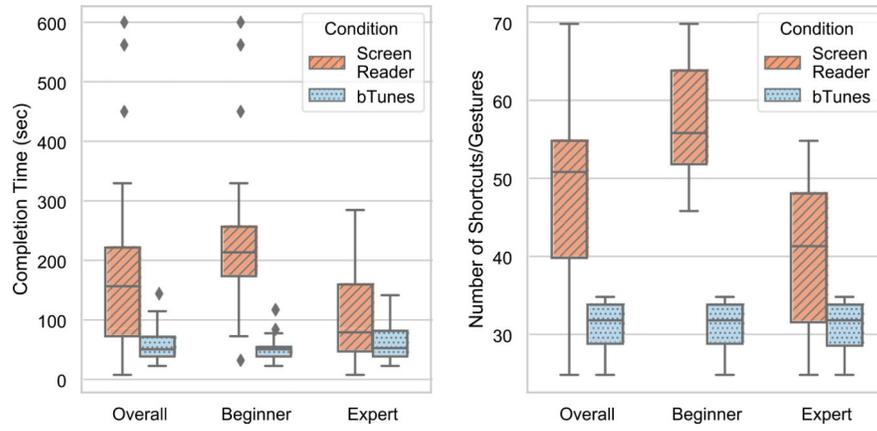
(b) Interaction workflow of bTunes.

**Fig. 1.**

Illustration of bTunes for Microsoft Word: (a) application ribbons containing multitude of commands that can easily accessed with a point-and-click mouse, but harder to access with a keyboard-based screen reader; (b) alternative rotate-and-press bTunes interface for non-visually accessing ribbon commands. Instead of shifting screen-reader focus from main edit area to the ribbon and then sequentially navigating the ribbons, the screen-reader user can simply press-and-hold the Dial to bring up a menu dashboard containing the outer ribbons (i.e., Home, Insert, etc.). The user can then rotate the Dial to focus on the desired ribbon, and then press to shift focus to the corresponding inner ribbon, specifically, bTunes opens up a dialog box with the corresponding inner ribbon commands. The user can repeat the same rotate and press gestures to select commands and command options (if any).



**Fig. 2.**  
An architectural overview of bTunes.



**Fig. 3.** Completion times and number of shortcuts/gestures for Task 1 under two study conditions, i.e., screen reader and bTunes.

**Table 1.**

Participant demographics. All information shown were self-reported by the participants in the study.

ID	Age/Gender	Screen reader	Word processor	Proficiency	Frequency
P1	39/M	JAWS, VoiceOver	Word, Pages	Expert	Daily
P2	54/M	JAWS	Word	Beginner	2 days a week
P3	46/F	JAWS, NVDA	Word, WordPad	Expert	Daily
P4	31/F	JAWS, NVDA	Word, Google Docs, WordPad	Expert	Daily
P5	60/M	JAWS	Word	Beginner	3 days a week
P6	61/F	JAWS	Word	Beginner	2 days a week
P7	44/M	JAWS, VoiceOver	Word, Pages	Expert	5 days a week
P8	45/M	JAWS, NVDA, System access	Word	Expert	5 days a week
P9	35/M	JAWS	Word	Beginner	2 days a week
P10	54/M	JAWS, VoiceOver	Word, Pages	Beginner	1day a week
P11	63/F	JAWS	Word	Expert	Daily
P12	32/F	JAWS, System access	Word, Google Docs	Expert	Daily
P13	56/M	JAWS	Word	Beginner	1day a week
P14	62/M	JAWS	Word	Beginner	2 days a week
P15	36/F	JAWS, NVDA	Word, WordPad	Expert	Daily