Tilt-Explore: Making Tilt Gestures Usable for Low-Vision Smartphone Users

Farhani Momotaz
Pennsylvania State University
University Park, Pennsylvania, USA
fbm5122@psu.edu

Syed Masum Billah
Pennsylvania State University
University Park, Pennsylvania, USA
sbillah@psu.edu

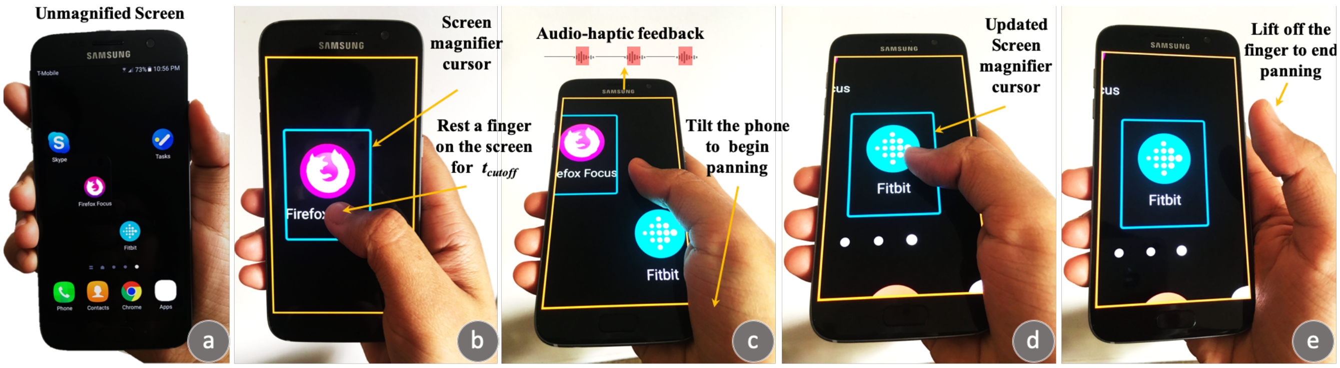


Figure 1: An illustration of Tilt-Explore technique to find an app (e.g., Fitbit) on the Home screen. (a) The unmagnified Home screen. (b) A user first rests a finger anywhere on the magnified screen and waits for a predefined time to activate panning. (c) Without lifting the finger, the user tilts the phone to a direction, and UI elements on that direction glide under their fingertip. (d) While panning, the UI element gliding under their fingertip automatically gets focused. (e) The user stops panning by lifting the finger off the screen. The last UI element under their fingertip remains focused.

ABSTRACT

People with low vision interact with smartphones using assistive technologies like screen magnifiers, which provide built-in touch gestures to pan and zoom onscreen content. These gestures are often cumbersome and require bimanual interaction. Of particular interest is panning gestures, which are issued frequently, which involve 2- or 3-finger dragging. This paper aims to utilize tilt-based interaction as a single-handed alternative to built-in panning gestures. To that end, we first identified our design space from the literature and conducted an exploratory user study with 12 low-vision participants to understand key challenges. Among many findings, the study revealed that built-in panning gestures are error-prone, and most tilt-based interaction techniques are designed for sighted users, which low vision users struggle to use as-is. We addressed these challenges by adapting low-vision users' interaction behavior and proposed Tilt-Explore, a new screen magnifier mode that enables tilt-to-pan. A second study with 16 low-vision participants revealed that, compared to built-in gestures, the participants were significantly less error-prone; and for lower magnification scale (e.g., <4x), they were significantly more efficient with Tilt-Explore. These findings indicate Tilt-Explore is a promising alternative to built-in panning gestures.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility systems and tools**; *User centered design.*

KEYWORDS

Smartphone; screen magnifier; tilting, panning; IMU, motion sensor; low-vision, vision impairments; tilt-exploration.

ACM Reference Format:
Farhani Momotaz and Syed Masum Billah. 2021. Tilt-Explore: Making Tilt Gestures Usable for Low-Vision Smartphone Users. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3472749.3474813

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10...$15.00
https://doi.org/10.1145/3472749.3474813

1 INTRODUCTION

Low-vision is broadly defined as a visual impairment that cannot be fully corrected even with glasses, medication, or surgery [6]. It encompasses loss of peripheral or central vision, blurred vision, extreme light sensitivity, tunnel vision, and near-total blindness. People with low vision rely on special-purpose assistive technology, mainly screen magnifiers, such as Zoom [5] in iPhone, Magnification in Android [2], to interact with smartphones.

Screen magnifiers in smartphones provide a set of touch gestures to let low-vision users zoom and pan onscreen content. Unfortunately, these gestures are designed by overloading the standard touch gestures, such as 1-finger touch or tap or drag. For instance, to pan, low-vision users need to drag the magnified screen with 3 fingers on iOS [5] or with 2 fingers on Android [3]. Figure 2.b shows these gestures.

Unsurprisingly, researchers have identified a number of usability issues with these overloaded gestures. *First*, these gestures are cumbersome to use [24]; and quickly become tiring due to repeated use [61]. *Second*, these gestures require bimanual interaction, i.e., holding the smartphone with one hand while doing the gestures with the other hand. Scenarios when another hand is encumbered [48], screen magnifiers could be challenging to use. *Third*, since all smartphone touch gestures involve some subset of finger combinations, it is easy to mix up one for the other.

This paper seeks to address these usability problems associated with built-in panning gestures in screen magnifiers. We believe that making panning gestures more usable is important because panning is done more extensively than zooming, which is portrayed by Furnas et al. [28] in their space-scale diagrams. In theory, zooming changes the magnification scale, whereas panning moves a fixed-size viewport in the magnified space. A 3D space-scale diagram can portray the zooming and panning relationship, where zooming represents moving along a 1D vertical axis, and panning represents moving a fixed-sized viewport over 2D horizontal planes at different scales extensively. As such, any improvement on panning gestures can substantially benefit low-vision users who rely on panning for everyday smartphone usage.

To that end, we explored tilt-based single-handed interaction that rely on smartphones' IMU sensors. Tilt-based interactions are widely used in mobile gaming [47, 63] because these gestures expand the input space of smartphones' built-in touch gestures. The combination of tilt-motion and touch has also been explored to enable single-handed interaction on mobile devices [25, 36, 65, 71]. Drawing on literature, we first identify the design space and parameters of tilt-based interaction for low-vision. Next, we refined our design space by conducting a study with 12 low-vision users.

Among many findings, our formative study revealed that panning gestures are error-prone, but their behavior is deterministic; low-vision users utilize both screen readers and screen magnifiers; not all low-vision users need single-handed panning, tilt-based panning could augment the touch-based built-in gestures, and existing techniques for tilt-based interaction are hard to use by low-vision users – they struggle to execute tilt delimiters, control tilt motion during panning, and stop panning when the target is seen in the viewport.

Informed by this study, we designed ***Tilt-Explore***, a usable tilt-to-pan mode for screen magnifiers. Tilt-Explore also incorporates low-vision users' preferences for tilt gesture delimiter, introduces features from screen readers, and maps tilt angles to panning distances through a custom transfer function that accommodates low-vision users' needs.

Tilt-Explore mode is illustrated in Figure 1. A low-vision user first rests a finger anywhere on the screen for 800ms, then *tilts* the phone to *explore* magnified content, as the content glides under their fingertip, it automatically gets focused. To stop panning, the user lifts their finger off the screen. To interact with the focused UI, they can double-tap anywhere on the screen.

A second user study with 16 low-vision screen magnifier users suggests that Tilt-Explore mode is as effective as the built-in panning gestures, easy-to-use, significantly less error-prone, and more engaging to low-vision users who are 45 or under.

We summarize our contributions as follows:

- A formative study revealing the usability issues with panning gestures and the challenges in tilt-based interaction for low-vision users.
- The design and development of Tilt-Explore, the first of its kind for low-vision accessibility.
- A summative study to evaluate the performance and user satisfaction of Tilt-Explore.

2 RELATED WORK

This section reviews related work on accessibility issues of screen magnifiers; general screen magnification techniques and their limitations; and topics, such as augmenting input space in smartphones using motion sensors, audio haptic feedback, and target acquisition.

2.1 Usability Issues with Screen Magnifiers

Accessibility problems with screen magnifiers have received attention from early on [16, 21, 22, 43]. Kline et al. [43], for instance, highlighted the locality issue caused by the magnifier's limited viewport and proposed a dual-mode magnifier: mobile and anchored. In mobile mode, the magnification window follows the cursor on the screen. In contrast, in the anchored mode, a fixed screen area is designated as the viewport, and the screen region surrounding the cursor, as it is moved around, is displayed in this viewport. They identified two essential features of screen magnifiers: support for indicating the cursor location with discernible visual markers; and support for customizing screen magnifiers' settings seamlessly. Fraser et al. [27], who reviewed the findings of prior research on assistive technologies, similarly reported the need for cursor enhancements for low-vision users. Zhao et al. [73] recommended configuring Kline's "mobile mode" to "full-screen", i.e., setting the magnification viewport to cover the entire screen, by default.

Besides identifying accessibility needs, researchers identified several usability challenges with screen magnifiers. For example, Theofanos et al. [64] highlighted the challenge of obtaining the gist of a webpage due to screen magnifiers' limited viewport. They also found that magnifying whitespace in a webpage disorients low-vision users. Other researchers also identified the indiscriminate magnification of onscreen content as the root cause of usability challenges in screen magnifiers [19, 34]. To that end, Billah et al. [19] proposed a differential magnification technique, which magnifies whitespace disproportionately to keep contextual elements in the viewport. Bigham [17] proposed to utilize the redundant space in a screen to amplify the text without causing "negative side effects", such as the magnified text overlapping with other screen objects and other content in the neighborhood. Agarwal et al. [7] proposed widget-specific magnification based on widgets' semantic properties in desktops.

Researchers who studied the impact of screen magnifiers on reading performance found no "one-size-fits-all" accessibility solution for the spectrum of eye conditions that low-vision entails. We observed similar findings in our studies. Hallett et al. [33] studied the impact on reading comprehension with and without word wrapping, and reported that screen magnifiers cause discomfort because of their lack of support for word wrapping.

Although originally reported for screen magnifiers in desktops, all of these usability challenges occur in screen magnifiers in smartphones. Even worst, screen magnifiers in smartphones pose two additional challenges: the complex multi-finger, multi-tap/-drag/-touch gestures for panning and zooming that are error-prone; and the excessive use of panning gestures, due to a smaller form factor of smartphones' screens, compared to desktops', is physically tiring [61]. Recently, researchers have addressed some of these challenges in non-smartphone devices, such as head-mounted display [60, 72]. In this paper, we address panning challenges exclusively on smartphone devices.

2.2 Input Space Augmentation with Tilt Gestures in Smartphones

Researchers have long been exploring ways to create new input modalities to expand the input space of smartphone gestures. They have leveraged the multi-touch capability of touchscreens, as well as the sensors [9] hosted in smartphones. For instance, Avery et al. [11] expanded the standard pan-and-zoom interaction by proposing a transient gesture that reduces the need for repeated transitions between multiple resolutions and locations. Likewise, the screen magnifier on iPhone (named Zoom) offers an onscreen, circular, and joystick-looking magnification control.

The use of motion sensors for interacting with virtual objects appeared in the work of Rekimoto [53] and Weberg et al. [67], where they used tilt motion to select menu items, interact with scroll-bars, pan, and zoom around a digital workspace. Harrison et al. [34], Small & Ishii [59], and Bartlett [14] extended the use of motion sensors for navigating widgets on mobile devices.

The combination of tilt-motion and touch have also been explored to enable single-handed interaction on mobile devices [25, 36, 65, 71]. For example, Yeo et al. [71] proposed gesture typing with one hand; Chang et al. [25] assisted targeting and extended thumb's reach; Tsandilas et al. [65] enhanced navigation with quick command gestures; and Hinckley [35, 36] proposed compound motion gestures, such as tilt-to-zoom, pivot-to-lock, and shake-to-delete. Our work draws on this large body of literature.

2.3 Triggers or Delimiters in Tilt Gestures

Ruiz et al. [56] demonstrated that smartphone users naturally associate the tilt motion gesture with panning, which is one of our motivations to investigate tilt gestures for panning in screen magnifiers for low-vision users.

Several smartphone apps [54, 58] utilize tilt gestures to pan (or scroll) static, non-interactive content (e.g., images). To turn on or off tilt actions, RotoView [54] uses a quick "throw" gesture as the trigger, whereas holding the screen with one finger serves as the trigger in Samsung Galaxy S4 [58]. Ruiz et al. [55] propose a "DoubleFlip" motion gesture as the trigger, where the user holds the phone with one hand and rotates it 180°along its long side and re-rotates it back to the original position. For a complete review on tilt gestures, refer to Teather et al. [62].

2.4 Audio-Haptic feedback

Audio-haptic cues have been used to assist blind and low-vision users in numerous projects. For example, these cues are used to reorient blind and low-vision users during content navigation [18, 19], wayfinding, and exploring maps [13, 51, 57]. Usually, prior work used a combination of synthesized speech, sonification, and haptic patterns to convey different information. We also use audio-haptic feedback to report ongoing panning status and announce whether the magnified viewport has reached screen boundaries.

2.5 Target Acquisition

People with low vision need to point and select targets on the screen with ease and efficiency. Toward this objective, several approaches have been proposed [15, 20, 30, 37, 40, 70]; the main idea being the modification of the presentation of targets or cursor or both. In contrast to these approaches that gauge improvements based on Fitts's law [44], Object Pointing [32] overrides the default cursor behavior to directly jump from one interface object to another, thereby bypassing the intervening white spaces altogether. Although it is effective, especially for people with motor impairments, overriding the default ad-hoc pointing behavior is not desirable for low-vision users, who struggle to keep themselves oriented in the magnified view. In this paper, we do not skip whitespace between two elements during panning, although a limited skipping may be feasible as suggested by Billah et al. [19].

3 DESIGN SPACE FOR TILT-BASED INTERACTION FOR LOW-VISION USERS

Drawing on the large body of tilt-based research [45, 52, 55, 56, 63], we first identified key factors influencing the performance of tilt inputs. For each factor, we then aimed to find optimal parameters that could suit the needs of low-vision users. In order to identify these needs, we explored literature on several related areas, including accessible non-visual gestural input [24, 41, 42, 66], tilt inputs [45, 62], tilt-touch inputs [25, 63, 65], and human dexterity [12, 31]. In the following sections, we describe our design space and initial design parameters.

3.1 Axial Range-of-Motion

The dexterity of human hand is very limited to rotate along Z axis (yaw) [29]. As such, we only considered rotation along X axis (pitch) and Y axis (roll). Furthermore, we limited the range of pitch and roll angle to ±30°, measured from the initial rest position, to ensure the visibility of screen [49] and user comfort [63]. Following the recommendation of Hinckley et al. [35], we set the dead band for tilting to ±3° to ignore small, unintended rotational movements.

3.2 Tilt Gesture Delimiter

In any tilt-based interaction, the system needs to distinguish an intended tilt motion from the unintended or accidental tilts [55]. Researchers proposed several motion gesture delimiters to indicate the beginning or ending of an intended tilt motion. These delimiters or triggers fall under two categories: (i) exaggerated movements or explicit mode switching, and (ii) implicit mode switching by combining motion and touch gestures.

3.2.1 Delimiters with Exaggerated Movements. We explored DoubleFlip [55] and Throw [54] motion gestures because these are resistant to false positive conditions and achieve a high recognition rate.

D1 `DoubleFlip motion gesture`: To turn *on*, a user rotates the phone along its long side so that the phone screen is away and then back [55]. To turn *off*, the user performs the gesture again, or remains inactive for a certain time to *time-out*.

D2 `Throw motion gesture`: to turn *on*, the user abruptly moves the phone back and forth by a small margin while facing the screen [54]. Similar to D1, to turn off, the user performs this gesture again or remains idle to time out.

3.2.2 Delimiters with Implicit Mode Switching. We considered the following two options:

D3 `Press-n-hold a clutch button`: the user touches (or presses) and holds a *clutch* button for the duration of tilt [63, 68]. The clutch could be a physical button (e.g., volume buttons of a smartphone) or a virtual button on the screen (e.g., a floating touch region similar to iPhones' AssistiveTouch [10] or Facebook Messenger's chat head). To turn *off*, the user releases the clutch button.

D4 `Use a toggle clutch button`: The user touches or presses a *clutch* button which toggles between on and off states [71].

3.3 Panning Control

To enable panning via tilt motion, the user tilts the phone along the roll or pitch axis, and the magnified viewport moves along the direction of tilt [56]. Prior work suggested that this movement can either be controlled by *velocity* or *position*. In velocity control, the movement is proportionate to the amount of tilt per second. In contrast, position control is described as "the farther the device is tilted, the farther from center the cursor is positioned" [62].

Oakley et al. and Kim et al. found that position control outperforms velocity control on bounded scrolling or pannings, such as menu selection [49] and tele-manipulation [69]; however, Teather et al. [62] reported that it is not suitable for unbounded panning. Since low-vision users apply a wide range of magnification scales —from 1× to 15×—we considered panning as an unbounded operation. Therefore, we opted for velocity control in our design.

Following the recommendation of Rahman et al. [52] for velocity control, we skipped discretizing the raw tilt angle into 14±1 finite controllable levels, which is suggested for position control by Partridge et al. [50] and Oakley et al. [49].

3.4 Transfer Function

Instead of setting a predefined "zero orientation", we chose the orientation when a user initiates tilting, allowing them to hold the device at their comfortable posture [35]. We refer the initial orientation as `static reference`, O_0^S. At any time t, the tilt angle ($\Delta\theta_t$) is computed as the difference between the current orientation (O_t) and the `static reference`, i.e., $\Delta\theta_t = O_t - O_0^S$. To translate tilt angle to panning speed, v_{t_x} (along X) and v_{t_y} (along Y), we used Hinckley's transfer function [35] with minor modification:

$$\begin{aligned} v_{t_{\{x,y\}}} &= K * screen\_dim_{\{x,y\}} * sgn(\Delta\theta_{t_{\{x,y\}}}) \\ &\quad * max(|\Delta\theta_{t_{\{x,y\}}}| - \theta_{min}, 0)^{\alpha} \end{aligned} \quad (1)$$

Here, K is the control gain, θ_{min} is the size of the dead band, α is the non-linear parameter, and $screen\_dim_{\{x,y\}}$ is the screen dimension in inches along X or Y axis. We set $K = 0.3$ and $\alpha = 1$ to keep the transfer function linear and manageable. Besides K, note that the panning speed depends on screen dimensions.

At any time t, we restricted panning along a single axis corresponding to the maximum of ($|\Delta\theta_{t_x}|, |\Delta\theta_{t_y}|$). The velocity of the other axis is set to 0. This restrictive 1D panning is found to be more usable than unrestricted 2D panning by Bartlett[14] and Hinckley et al. [35].

3.5 Audio-Haptic Feedback

Tilt input does not offer tactile feedback. As such, we chose to provide simple audio-haptic feedback (e.g., a short beep and a weak vibration) to indicate the start and end of a tilt gesture. We also provided special audio-haptic feedback with a long beep and stronger vibration to announce boundary conditions, as recommended by [18]. Boundary conditions occur when the tilt angle exceeds ±30°, or the magnified viewport hits an edge of the screen.

4 FORMATIVE STUDY

We conducted an IRB-approved study to understand feasibility of using tilt-based panning for people with low vision who use screen magnifiers on smartphones. We aimed at collecting user feedback and preferences on different design parameters identified in Section 3. We also observed how participants react to individual design parameters and whether our initial prototype needed further refinement.

4.1 Participants

We recruited 12 low-vision participants (7 males, 5 females) through local mailing lists, university mailing lists, and public posts on Facebook groups. Our inclusion criteria included low-vision adults who need screen magnifiers on their smartphones. The participants varied in age from 29 to 77 ($M = 53.8$, $SD = 16.81$) and professions: *IT instructors* = 3, *artist* = 1, *attorney service representative* = 1, *cashier* = 1, *stock broker* = 1, *high-school teachers* = 2, *factory worker* = 1, and 2 were never employed. Table 1 presents their demographics. All participants were aware of their diagnosed eye condition; however, some were unsure about their precise visual acuity. They lived in the New York metropolitan area, including New Jersey and Connecticut.

4.2 Apparatus: Tilt-to-Pan Prototype

To realize our design parameters identified in Section 3, we developed a prototype on top of Android's builtin magnifier [2], which was deployed as an accessibility service. This prototype supported tilt-to-pan gesture, as well as Android's and iPhone's builtin panning gestures (e.g., 2-finger drag and 3-finger drag, respectively). When tilting starts, the prototype system computes the device orientation (O_0^S) using `getRotationMatrix` and `getOrientation`

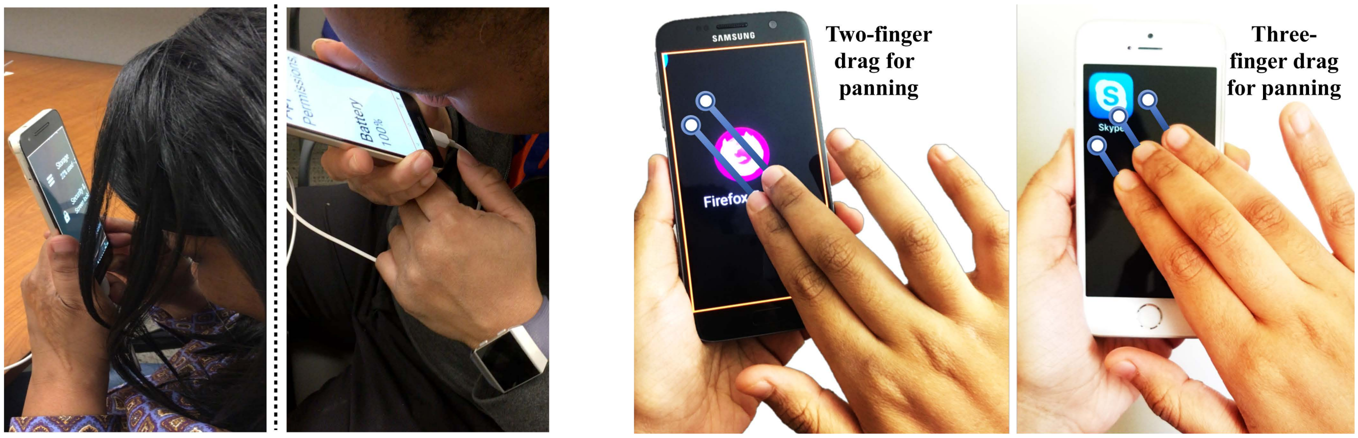


Figure 2: (a) Some low-vision participants brought the phone close to their eyes to reduce panning effort during our formative study; (b) Illustrations of screen magnifiers' built-in panning gestures: 2-finger swipe/drag on Android (left) and 3-finger swipe/drag on iPhone (right).

Table 1: Participant demographics in the formative study. Note that participants' visual acuities are self-reported and may not be accurate.

ID	Age/ Sex	Diagnosis (C: Congenital, A: Adventitious)	Visual Acuity (L: Left, R:Right)	Smartphone Used	Expertise
P1	71/F	Glaucoma (A)	L: 20/200; R: 0	iPhone 8	Intermediate
P2	48/M	Macular Telangiectasia Type 2 (A)	L: 20/100; R: 20/300	Android	Intermediate
P3	34/M	Albinism (C)	L: 20/200; R: 20/100	iPhone 4	Expert
P4	47/M	Optic atrophy (A)	L: 20/100; R:20/200	Android	Intermediate
P5	35/M	Optic atrophy, Retinitis pigmentosa (A)	L: 20/700; R: 0	iPhone 6s	Expert
P6	62/F	Congenital retinal scar (C)	L: 20/400; R: 20/400	iPhone 6s plus	Beginner
P7	29/F	Early onset Glaucoma (C)	L: 0; R: 20/400	iPhone 10	Expert
P8	53/M	Retinal degeneration (A)	Unknown, but good	Feature phone, Android	Beginner
P9	45/F	Leber's Congenital Amaurosis (C)	L: 20/200; R: 20/400	Android, HTC	Intermediate
P10	77/M	Glaucoma (A)	L: Unknown; R: 0	Feature phone, Android	Beginner
P11	71/F	Diabetic retinopathy (A)	Unknown, but good	Android (Galaxy S7)	Intermediate
P12	74/M	Strabismus & cataract (C)	20/500 (both)	Feature phone, Android	Beginner

functions in Android Manager [8], and samples sensor data from accelerometer and gyroscope at a rate of $f_s = 50Hz$. For delimiters D3 and D4, we created a floating clutch button on the screen by implementing a background service that adds a floating widget into the view hierarchy of the current screen. We also overrode Android's volume buttons to make them behave as clutch buttons.

4.3 Study Design

We adopted a concurrent think-aloud protocol. Our primary objective was to collect user feedback, preferences, and observe how participants react to our prototype. The study had two parts as described below:

4.3.1 Part 1: Feedback on Tilt Delimiters. The participants executed tilt delimiters, D1, D2, D3, and D4, five times each. These delimiters are described in Section 3.2. D3 and D4 had two options for clutching: (i) an onscreen floating button or (ii) physical volume buttons.

4.3.2 Part 2: Feedback on Tilt-to-Pan Gesture. Next, the participants performed two visual tasks (described below) related to visual search and exploration in 2D, and reading rows in 1D layouts. We believe these tasks represent most real-world interactions in smartphones because app development frameworks in Android [46] and iOS [38] promote organizing UI elements and content in 2D grids (e.g., Home screen, Calendar, Maps) or in 1D lists (e.g., Settings, recent calls, Facebook feed). Each task had 5 trials and 2 panning conditions: (i) built-in gestures and (ii) tilt-to-pan prototype.

- **T1** *Visual Search and Exploration in 2D Grids*: Given its name, find an app on the Home screen. For example, find `Fitbit` app in Figure 1.
- **T2** *Read a row in 1D Lists*: Read the value of a given attribute in Android's `Settings` window. For example, read the row named `Battery` in `Settings` and report the current battery level.

4.4 Study Procedure

The study was conducted on a 5.7" Android smartphone (model: Google Nexus 6P, API v.27) with 1440×2560 resolution, in an office environment, before the COVID-19 pandemic. After consent, we asked participants to introduce themselves, their educational and professional background, history of visual impairment, and their use of assistive technologies. We then asked their preferred magnification settings, such as contrasts, magnification scale, and showed them how to apply these settings on the study phone. After demonstrating supported gestures related to magnification, panning, and tilting, we let them practice and familiarize themselves with the study phone for ∼15 minutes. They were allowed to use their familiar gestures to change the magnifier's scale. Those who needed assistance to make tilt gestures or tilt delimiters for the first time, we held their hand and performed the gesture until they became comfortable making their own.

The participants were instructed to voice their thoughts during a task. Each session was audio-video recorded, conducted by two experimenters, who observed how the participants performed a task and took notes during a session. The orders and conditions in each task were counterbalanced. All conversations were in English. Each session lasted for 2 hours. We compensated participants with an hourly rate of USD $25. Next, we describe procedures specific to each part of the study.

4.4.1 Procedure in Part 1. We asked participants to magnify the screen to a comfortable scale. When executing each delimiter, they were instructed to trigger the delimiter on, tilt the phone for some time (e.g., ∼5s), then trigger it off. When using the floating clutch button in D3 and D4, they were additionally instructed to position the floating button at a comfortable location on the screen. The correctness of each execution was at the discretion of the experimenters. They also asked questions about participants' preferences and perceived challenges for each delimiter in the end. In total, 30 trials were observed in Part 1.

4.4.2 Procedure in Part 2. In this part, we let the participants use their preferred delimiter reported in part 1. For each trial in T1, the experimenters randomly placed 10 apps on the home screen, magnified the screen to an individual's need, and ensured that the target was not in the current viewport to warrant panning. A trial in T1 was completed when the participants reported seeing the target within the current magnified viewport. Similarly, for each trial in T2, the experimenters chose a different attribute from different sub-windows of `Settings`. A trial was completed when the participants read out the value of the prompted attribute. Additionally, while panning with built-in gestures, they were allowed to benefit from their prior experience using screen magnifiers. However, they were instructed not to use built-in panning gestures during tilt-to-pan condition. The experimenters allocated 3 minutes for each trial, observed 20 trials in total, and in the end, asked for participants' feedback and comment on a task and study condition.

4.5 Findings

We analyzed experimenters' notes, audio transcripts, and recorded video to understand user behavior, strategies, challenges, and preferences for panning and tilting gestures. Note that we did not report task completion times because such measures are not reliable in a think-aloud study. The following sections describe our key findings.

4.5.1 Preference for Tilt Delimiter. All participants struggled to make exaggerated movements required for *D1* (`DoubleFlip`) and *D2* (`Throw`). They consistently rated these delimiters the lowest. P8 and P12 dropped the phone once. Besides physical struggle, 8 out of 12 participants mentioned a major drawback with these delimiters — they were unable to keep track of their last visual location on the screen after performing these gestures. P3 explained: "*When I flip over the phone and flip it back, or shake the phone, I lost track of where I was looking at*". These participants stated that they would be reluctant to use a technique that requires exaggerated movements. As such, we eliminated D1 and D2 from our design space.

Between D3 (`Press-n-hold a clutch`) and D4 (`Toggle clutch`), they preferred D3 because it required "less work", and panning could be stopped "instantaneously" by lifting a finger off the clutch, which "makes sense"; whereas D4 needed an "extra" step, such as touching or pressing the clutch button again, to end panning. P6 elaborated: "*... my eyes are slow but in this case, I saw the target. By the time I touch the clutch, the target is gone [off-screen]*". 3 participants reported another issue with D4 – they "forgot" to tap on the clutch in the end. Therefore, we removed D4 from our design space.

The participants expressed concerns about both clutch options. They strongly opposed the use of volume buttons. Some commented that pressing a volume button with a thumb while tilting the phone was not convenient, as it limited their wrist movement. The experimenters also observed difficulties pressing a volume button while tilting. P7 provided more insightful feedback: *"I use VoiceOver [screen reader] a lot... Volume buttons are critical to me, and I'd not mess around with them"*. Based on these comments, we removed volume buttons from our design space.

Regarding the last remaining option, a floating clutch button, most participants struggled to discern it on the screen due to their low visual acuity. They mentioned that they must put an "effort" to locate it each time they used tilt-to-pan. Again, P7 described her experience as follows: *"... your floating button is very similar to assistive touch on my iPhone. I do not use assistive touch because it is so hard to see"*. Both P3 and P7 questioned the necessity of having a separate clutch button at all. Instead, they suggested touching anywhere on the screen. We note that touching anywhere on the screen without performing a tap action is not trivial. Nonetheless, we conclude that our initial design had room for improvement.

4.5.2 Use Cases for Tilt-to-Pan. All participants mentioned that built-in panning gestures are more deterministic than the proposed tilt-to-pan. If they sensed that the target is nearby or noticed parts of the target in the viewport, they would prefer built-in panning gestures. But if the target is afar, they would prefer tilt-to-pan to explore magnified content quickly with less effort, i.e., less "finger dragging" (P1). Thus, we found that the participants considered tilt-to-pan as an addition rather than a substitution.

We observed another use case for tilt-to-pan. In task T1, 6 participants, in an attempt to reduce panning effort, applied a lower magnification scale and brought the phone very close to their eyes (see Figure 2.a). Doing so left little space between the screen and their eyes, making it challenging for them to issue built-in pan gestures. These participants repeated the same pattern when using tilt-to-pan – they brought the phone close to their eyes, resting it on the palm of their non-dominant hand, and controlled the direction and amount of tilt with their dominant hand. They explained that they are accustomed to bringing their phones close to their eyes, which they are unwilling to change. Upon further inquiry, they mentioned that although a single-handed interaction is not a priority for them, they found panning via tilting more accommodating to their needs.

In sum, we concluded that the preference for panning gestures is context-dependent, and the need for an additional panning gesture is well substantiated.

4.5.3 Chaining Tilt Motion. The participants who needed higher magnification (e.g., scale > 5×) experienced disorientation in both study conditions because the panning speed became too fast for their eyes to process. Most participants raised a concern about our prototype that the panning speed is very sensitive to tilt motion and needed to be "tamed". Surprisingly, 4 participants adopted an interesting strategy: they tilted the phone along an axis quickly "to create a mental scope" and gradually tilted the phone backward, causing panning to pause momentarily before picking up speed in the opposite direction. They mentioned that the momentary pause of panning is important, and it "buys some time" for their eyes to process visual content. They repeated the same pattern throughout panning as if they chained a wave of small tilt motions. This was a key observation that led us to redesign a new transfer function for low-vision users.

4.5.4 Panning Gestures are Error Prone. We observed that when participants were panning with 3-finger–dragging, sometimes not all three fingers touched the screen simultaneously. The finger that touches the screen right before others registers an "unintended" one-finger tap event, which accidentally opens a different window. We further observed that the recovery time from such accidental context switches varied widely, ranging from 5s to 90s.

Coincidentally, we observed similar patterns with our prototype – when participants were close to the target, they unintendedly touched the screen and switched to a different window—recovering from accidental context switches impacted their task completion time substantially. When asked, the participants mentioned that "unintended" taps negatively impacted their user experience. P3 shared his strategy in this regard: *"Sometimes, I take screenshots of an application which are saved in Photos... I then look at those shots using pinch-to-zoom gesture"*. Overall, they expressed hopelessness and blamed themselves for being "sloppy" or "too excited".

4.5.5 Simultaneous Use of Screen Reader and Screen Magnifier. We observed that in tasks that involved reading texts (e.g., task T2), most participants (except for P2 and P6) used a screen reader to supplement the screen magnifier. They mentioned that they would prefer screen readers for reading texts. They also stated that using a screen reader and a screen magnifier simultaneously is cumbersome because the gestures are different. However, they had to do it sometimes. They strongly recommended integrating basic screen reading features into our prototype.

4.5.6 Usability Issues with Tilt-to-Pan Prototype. We noticed that due to the limited axial range of motion (e.g., ±30°), the participants could not pan through the entire magnified screen in one continuous tilt. However, some participants kept on tilting beyond ±30° and rotated their head along at the direction of tilt to see the screen. It was an unexpected behavior, and we realized the need to devise a mechanism that could inform users when to stop tilting.

Some participants reported that they were confused when panning empty spaces, such as a solid background and whitespace, because the viewport seems unchanged. They recommended providing audio-haptic feedback at regular intervals during panning for confirmation.

We also noticed a concern regarding panning direction in response to tilt. The participants were split on moving the content in the direction of the tilt vs. the opposite direction of tilt. Thus, we added a personalized setting to our prototype.

4.5.7 Preferred Gestures to Change Scale. The participants were allowed to use their familiar screen magnifiers' gestures to change magnifiers' scale. Interestingly, after learning that our prototype supports 2-finger pinch-to-zoom, which they used to magnify webpages and images, most participants (except for P5) preferred it over other built-in gestures. We note that, unlike iPhone's default screen magnifier, pinch-to-zoom is built-in to screen magnifiers in Android. Besides pinch-to-zoom, some participants (P2, P4, and P11) toggled between magnified and unmagnified views by issuing either a 3-finger-double-tap or 1-finger-triple-tap gesture. In sum, we report that the participants preferred 2-finger pinch-to-zoom the most to change magnifiers' scale.

5 DESIGNING TILT-EXPLORE

The findings from our formative study revealed the most appropriate design choices for a usable tilt-to-pan interaction for low vision users. These were incorporated in our revised prototype, ***Tilt-Explore***, which we describe next.

5.1 Revised Tilt Gesture Delimiter

We learned that low-vision users prefer touching and holding a clutch button (i.e., D3) on the screen, but locating a floating clutch button on the screen could the challenging. Additionally, touching anywhere on the screen is preferred. We also learned that an unintended touch during panning may cause an accidental context switching, which needs to be prevented. Furthermore, incorporating basic screen reading features during tilt-to-panning is strongly recommended.

To satisfy these design goals, we borrowed a feature from screen readers, *Touch-Exploration*, commonly known as risk-free-exploration in the literature [41]. The Touch-Exploration mode is designed to make touch screen accessible for non-visual interaction.

In this mode, touching the screen only selects and highlights a UI element under the touch-point. It does not register a click or tap operation unless the user taps twice (anywhere) on the screen. We set a flag, FLAG_REQUEST_TOUCH_EXPLORATION_MODE [1] in Android Accessibility, to enable Touch-Exploration mode in our prototype. Informed by the formative study, we believe most low-vision users are familiar with Touch Exploration, as they often supplement screen magnifiers with screen readers. Incorporating Touch Exploration in our prototype would prevent accidental context switches, and enable touching and holding the screen with 1-finger for clutching.

5.2 Revised Transfer Function

We learned from Section 4.5.3 that low-vision users prefer panning speed to slow down in regular intervals to "buy some time" so that their eyes could process visual content in the current viewport. To satisfy this requirement, we revised the measurement of tilt angle $\Delta\theta_t$ in Section 3.4 as follows:

We first introduce a `dynamic reference` orientation $O^D_{[t/\lambda]}$ that updates in every λ seconds to capture the device's orientation at that moment. Here, $[t/\lambda]$ is the interval index, which is equivalent to time at $\lambda * \lfloor t/\lambda \rfloor$ seconds. Mathematically, $O^D_{[t/\lambda]} = O_{\lambda * \lfloor t/\lambda \rfloor}$. An illustration is shown in Figure 3.C.

Next, we capture the recent tilt angle $\Delta\theta^D_t$ with respect to $O^D_{[t/\lambda]}$. If we redefine the tilt angle captured in our initial prototype as $\Delta\theta^S_t$, i.e., tilt angle with respect to O^S_0, we can combine both $\Delta\theta^S_t$ and $\Delta\theta^D_t$ to compute the refined $\Delta\theta_t$. Equation 2 shows these measures. Finally, we compute panning speed v_{t_x} and v_{t_y} from $\Delta\theta_t$ using the same equation 1.

$$\begin{aligned} \Delta\theta^S_t &= O_t - O^S_0 \\ \Delta\theta^D_t &= O_t - O^D_{[t/\lambda]} \\ \Delta\theta_t &= (1-\eta) * \Delta\theta^S_t + \eta * \Delta\theta^D_t \end{aligned} \quad (2)$$

Here, λ represents the interval of panning to slow down, and η indicates responsiveness of our transfer function. We set $\lambda = 5s$ and $\eta = 0.8$, empirically.

From the perspective of low-vision users, they would observe that the panning speed slows down in every λ seconds. Figure 4 demonstrates how the refined $\Delta\theta_t$ in Tilt-Explore varies from the original $\Delta\theta^S_t$ in different traces of simulated tilt motion. The top-left trace shows that $\Delta\theta_t$ in Tilt-Explore drops in every $5s$ to slow down panning even though the user keeps on tilting in a same direction. Other traces show the resilience of our transfer function against abrupt tilting.

5.3 Revised Audio-Haptic feedback

Tilt-Explore provides feedback in synthesized audio, earcon, and vibrations (amplitude: default, duration: 10ms – 300ms). If the user hits a boundary, Tilt-Explore announces the name of the boundary, e.g. "left edge" along with strong haptic feedback (duration: $300ms$). Being aware of this information, the user may decide to stop panning by lifting their finger off the screen or tilt the phone at the opposite angle.

Tilt-Explore conveys its update of `dynamic reference` in every λ seconds via a beep earcon and medium-duration vibration. To inform users that the panning is active, Tilt-Explore provides a short beep at a user-defined interval. Lastly, Tilt-Explore informs users to lift their finger off at an extreme angle. The user can configure all feedback from Tilt-Explore's settings.

5.4 Revised Tilt-to-Pan Interaction

Since our prototype was considered additive rather than substitutive, we keep tilt-based interaction consistent with low-vision users' regular interaction paradigms. To that goal, we hope their regular interaction, such as double-tapping, swiping, would not take more than 800ms (empirically chosen). As such, we require users to rest one figure, preferably their thumb, on the screen for a longer duration (e.g., 800ms) to trigger tilt motion (see Figure 3.A). The system also provides distinctive audio-haptic feedback to indicate tilt-to-pan is enabled. Users can configure the panning to follow the direction of tilt or the opposite direction. While panning, a UI element glides under their fingertip and automatically gets focused (Figure 1). The user stops panning at any time by simply lifting their finger off the screen; however, the last UI element under their fingertip remains focused. To perform a click or tap operation on the focused UI, the user taps twice (anywhere) on the screen. At this point, the users can choose to use any touch-based built-in gestures to complete their tasks.

6 EVALUATION OF TILT-EXPLORE

We hypothesized that the performance of Tilt-Explore would be as good as the built-in panning gestures, if not better. In addition, Tilt-Explore would reduce the number of accidental taps, which usually cause unintended context switches during panning.

To that end, we conducted another study with 16 low-vision participants (10 males, 6 females). Their average age was 50.75 (median:50.5, SD:15.86, range: 30 – 71). We recruited them from the same channels with the same inclusion criteria as in the formative study. Table 2 presents their demographics. Two studies were 3-months apart. 4 participants (P1, P3, P8, and P11), who took part in the formative study, had little to no recollection of that study.

6.1 Apparatus

We learned from the formative study that unintended context switching caused a net increment of the task completion time. Furthermore, reproducing unintended context switches is hard. Therefore, we developed a *control app* that eliminates this uncertainly and makes our experiment reproducible. The app does not switch context when accidental taps happen. Instead, it records the number of times such incidents occurred. Figure 5 shows two study tasks on this app. Both Tilt-Explore and the control app are deployed on the same study phone used in our formative study.

6.2 Study Design

We used a within-subject design – all participants performed two visual tasks, related to visual search and exploration in 2D and reading text rows in 1D, using two study conditions, Tilt-Explore and built-in gestures.

T1 *Visual Search and Exploration in 2D*: Find a target UI element on the magnified screen, given its name and visual cues. For example, find a book cover titled "Hand Hand Fingers Thumb",

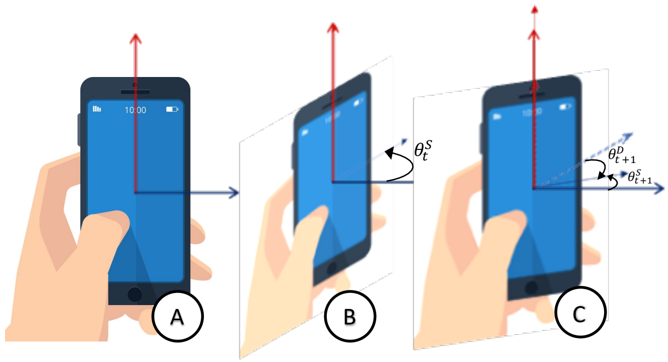


Figure 3: Illustrations of revised tilt delimiter and transfer function in Tilt-Explore. (A) a user rests their thumb (or any finger) on the screen for some time (e.g., $800ms$) to activate tilt-to-pan; (B) $\Delta\theta_t^S$ represents the amount of tilt at time t with respect to `static reference`; and (C) at time $t+1$, the user tilts the phone along the same axis but oppositely. Assuming that the `dynamic reference` was set at time t, $\Delta\theta_{t+1}^S$ and $\Delta\theta_{t+1}^D$ represent the tilt angles with respect to `static` and `dynamic` references.

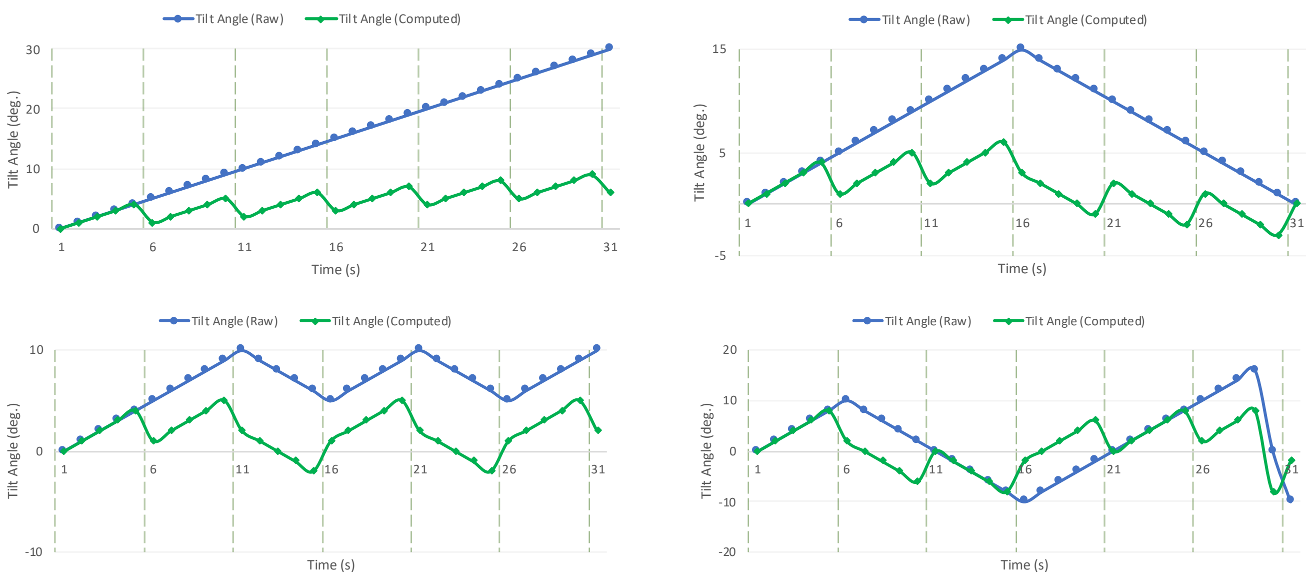


Figure 4: Characteristics of Tilt-Explore's transfer function in four different traces of tilt motion. Blue lines show tilt angles with respect to `static reference`, whereas green lines show the computed tilt angles used by Tilt-Explore. Each graph shows a trace of simulated tilt motion for $30s$. The top-left figure shows a trace of continuously tilting the phone clockwise along an axis. Notice that the computed tilt angle (green line) drops in every $5s$ to slow down panning even though the raw tilt angle (blue line) is increasing. Other figures show the resilience of our transfer function against abrupt tilting.

by Dr. Seuss, with yellow background, as shown in Figure 5.(a)-(b).

T2 *Read rows in 1D*: Read a short article aloud. An example is shown in Figure 5.(c)-(d).

Each task had the following 2 panning conditions:

C1 *Baseline*: Participants must use the built-in panning gestures in screen magnifiers. They were not allowed to use screen readers.

C1 *Tilt-Explore*: Participants must use tilt-to-pan. Use of built-in panning gestures or screen readers was not allowed.

T1 had 5 trials, 4 minutes each; and T1 had 1 trial, since a reading an article could take 5 to 10 min.

6.3 Study Setup and Procedure

We followed a similar procedure as in the formative study. The participants were given sufficient time (~10 min) and instructions to familiarize themselves with the study phone and conditions. The ordering of tasks and study conditions were counterbalanced. If a participant failed to complete any trial within the stipulated time limit, it was recorded as incomplete.

Table 2: Participant demographics in the Tilt-Explore study.

ID	Age/ Sex	Diagnosis (**C**: Congenital, **A**: Adventitious)	Smartphone Used
Q1	62/F	Macular Degeneration (C)	iPhone
Q2	48/M	Retinitis Pigmentosa (RP) (A)	iPhone
Q3	34/M	Albinism (C)	iPhone
Q4	63/M	Congenital Cataract (C)	iPhone
Q5	68/F	glaucoma (A)	iPhone
Q6	71/F	glaucoma (A)	iPhone
Q7	31/F	Retrograde optic atrophy (A)	iPhone
Q8	53/M	Retinal degeneration (A)	Android
Q9	46/M	Leber's congenital amaurosis (C)	iPhone
Q10	58/M	Stevens-Johnson syndrome(A)	iPhone
Q11	71/F	Retinitis Pigmentosa (C)	Android
Q12	42/M	Optic Atrophy (A)	Android
Q13	71/F	Diabetic Retinopathy (A)	iPhone
Q14	33/M	Corneal Erosion (A)	iPhone
Q15	31/M	Extremely Blurry Vision (A)	Android
Q16	30/M	Pathological Myopia (A)	Android

To gauge the usability of Tilt-Explore, we administered a post-completion System Usability Scale (SUS) questionnaire [23], which consists of 10 Likert scale statements, where the participant rated each statement on a scale of 1 – strongly disagree to 5 – strongly agree. Finally, we also administered NASA-TLX verbally to measure an individual's perceived workload. Each session lasted for 90 minutes. Procedures specific to a task are described below:

6.3.1 For task T1. Our control app displays a 4x4 grid. Each grid-cell represents a book, consisting of its cover art (dimension 24dp x 24dp), title (boldfaced, wrapped, 12-pt font), and author (12-pt font). Out of 16 grid cells, we randomly chose 10 cells to represent actual books and the remaining 6 cells to create whitespace or "blank". Each time, the app randomly draws 10 books from a sample of 20 books. Figure 5.a-b show this setup.

A trial in T1 started with a pop-up dialog asking the user to search for a book by reading out its title, author, and visual cues. When the user taps on the OK button, the system automatically sets the magnification to a higher scale (e.g., 8X). It randomly places the center of the current magnified viewport to at least three city blocks away from the target. Tapping on a book other than the target had no effect. The participants were instructed to adjust the magnification scale to their comfort level. A trial was considered complete when the user found the target book by double-tapping on it.

6.3.2 For task T2. For each trial, the control app randomly chose a short article from a sample of 15 articles. Figure 5.c-d show this setup. Each article has 12 lines on average (min: 10, max: 14). T2 represents many real-world tasks, such as reading news, product reviews, and notes. A trial was considered complete when the user read out the entire article.

6.4 Data Collection and Analysis

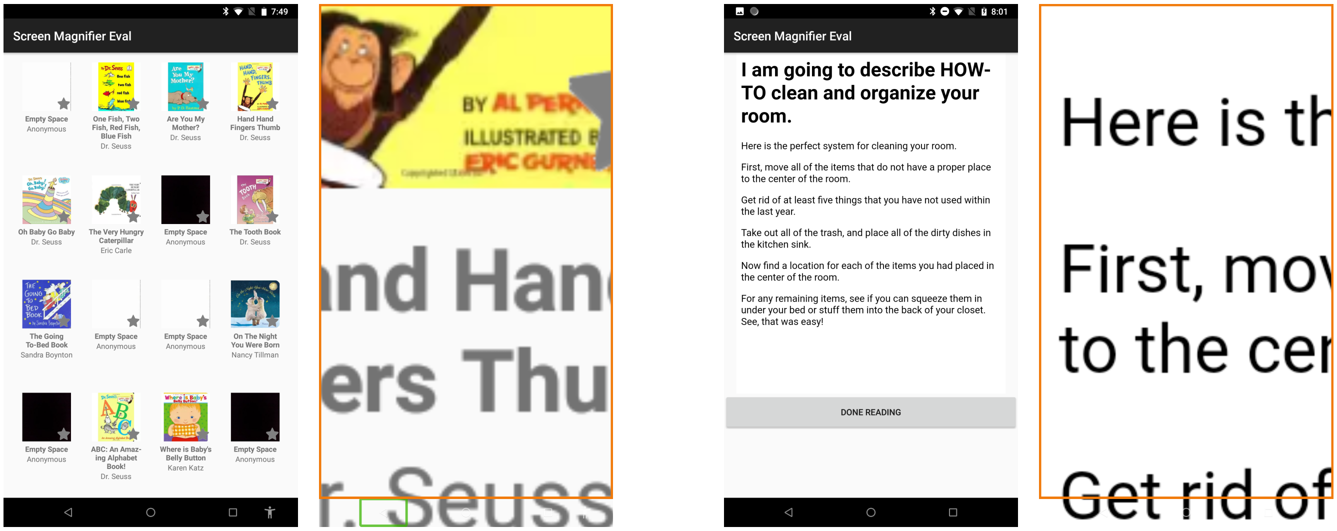
The study phone logged all user interactions during a task. We measured the number of panning gestures used, time to complete a task (in seconds), and the number of accidental taps during panning (T1). For the reading task (T2), we also computed the reading speed measured in words-per-minute (WPM), i.e., the total number of words divided by the task completion time. To analyze data, we first used Shapiro-Wilk to test normality. For non-normal data, we used Wilcoxon Signed Rank tests, paired t-tests otherwise.

6.5 Results

6.5.1 Panning Gesture Counts in Visual Search. Counting gestures in the baseline was simple: we counted the total number of times a user issued 2- or 3-finger-drag gestures to find the target. However, this measure was not well-defined for Tilt-Explore. So, we coded a gesture in Tilt-Explore as a user resting their finger on the screen, tilting the phone, until they lift the finger off the screen. With this coding, the participants used 79% less panning gestures with Tilt-Explore (Mean: 19.83, SD: 21.18) compared to baseline (Mean: 95.77, SD: 70.67) in task T1. A Wilcoxon Signed-Rank test showed a significant effect of the conditions on the panning-gesture count ($Z = -3.516, p < .001$). This was not surprising but reassuring; participants explored more content in the magnified view in one go with Tilt-Explore, as designed.

6.5.2 Accidental Taps during Visual Searching. Our control app recorded how many times a participant accidentally tapped on a UI that could cause an unintended context switch in practice. They accidentally tapped 3.72 times on average (SD: 4.17) with the baseline, compared to 0.15 times (SD: .20) with Tilt-Explore. A Wilcoxon Signed-rank test showed that this effect was significant ($Z = -3.411, p = .001$), which was expected.

What was not expected, though, was *why 0.15 times*? We expected Tilt-Explore to eliminate accidental taps since we used Touch-Exploration mode. Upon further inspection, we noticed that participants who were heavy-handed made these taps. These participants accidentally issued a "Split-Tap" or a "second-finger tap" gesture, originally proposed by Kane et al. [41], with their second



(a) (b) (c) (d)

Figure 5: Examples of 2 tasks: (a)-(b) a visual search task: find a book cover by its name and visual cues – (a) unmagnified view, (b) magnified view; (c)-(d) a reading task: read a short article aloud – (c) unmagnified view, (d) magnified view.

finger while their first finger (e.g., thumb) was on the screen, focusing on a UI. This was an unwanted side effect that highlights the challenges in designing assistive technologies. Finally, we note that iPhone users can disable the Split-Tap gesture from the Settings.

6.5.3 Completion Time in Visual Search. Participants took 131.17s. on average (SD: 123.16s) with baseline, and 134.85s (SD: 83.25s) with Tilt-Explore, which was not found to be statistically significant by a Wilcoxon Signed-Rank test. Although it validates our initial hypothesis that the panning performance in Tilt-Explore would be comparable to built-in gestures, these numbers do not capture the extent of Tilt-Explore's benefit.

Recall that the task completion time for low-vision users comprises two parts: time spent on ad-hoc panning, plus time to recover from unintended context switches. Note that our control app blocks unintended context switches, which could range from 5s to 90s, as described in Section 4.5.4. Since participants made significantly more accidental taps with the baseline, each of which could cause an unintended context switch, we believe Tilt-Explore would outperform the baseline in practice.

Interestingly, we observed that out of 16 participants, 6 who needed a smaller magnification scale (e.g., scale < 4×) took significantly less time to complete the task with Tilt-Explore, compared to the baseline ($Z = -2.201, p = .028$).

6.5.4 Reading Speed. The average reading speeds (WPM) were 31.60 (SD: 12.74) with baseline and 28.17 (SD: 12.47) with Tilt-Explore. Although participants were 10% slower with Tilt-Explore, this difference was not found to be significant, $t(10) = .952, p < .364$. This result also validated our initial hypothesis that Tilt-Explore would be comparable to built-in gestures.

However, the participants mentioned that they would certainly use a screen reader for Task T2, if they were allowed. We note that the average reading speed for people without vision impairment is 200 to 250 WPM [4]. In contrast, we recorded the maximum speed of ~44WPM in our study. Thus, we conclude that panning is not preferred for reading texts on smartphones for low-vision users.

6.6 Subjective Evaluation

6.6.1 User Feedback. All participants, except Q1, Q5, and Q11, rated Tilt-Explore highly favorably (5/5). 3 participants who rated unfavorably (2/5) had difficulty tilting the phone due to age-related conditions. A few participants raised concerns about using Tilt-Explore over an extended period. In contrast, some other participants mentioned that they had difficulty panning with the default touch gestures due to stiffness in their fingers. For them, Tilt-Explore was more convenient to use.

6.6.2 SUS Scores. We computed an overall composite score between 0 to 100 from participants' individual SUS scores as a measure for representing usability. The average SUS score for Tilt-Explore was 74.53 (SD: 26.14), which was significantly higher than that of the baseline (Mean: 50.15, SD: 23.42). Some participants recognized Tilt-Explore as a *hybrid* of screen magnifiers and screen readers, which we do not disagree. One participant said, *"Tilt-Explore feels like a screen reader and a screen magnifier on steroids."*

6.6.3 NASA TLX Scores. Figure 6 shows NASA-TLX scores of baseline and Tilt-Explore for (i) all participants, (ii) participants who were 45 or under, and (iii) participants who were over 45. There was no statistically significant difference between NASA-TLX scores in two conditions for all participants. Out of six measures in NASA-TLX, Wilcoxon Signed Ranks indicated that physical and temporal workload increased significantly with Tilt-Explore ($Z = 2.1, p = .012$ for physical, and $Z = -3.526, p = 0$ for temporal), and frustration reduced significantly ($Z = -3.10, p = .102$). This finding is consistent with participants's comments that we traded "frustration" with "temporal" demand.

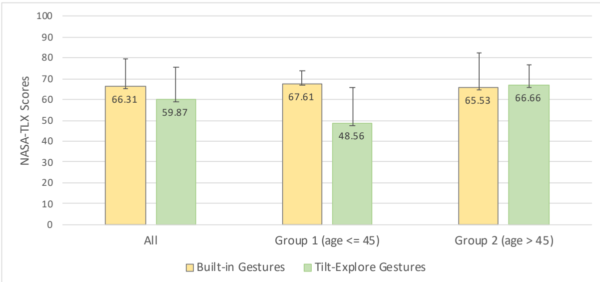


Figure 6: Comparisons of NASA-TLX scores between baseline and Tilt-Explore for all participants (left), for participants who were 45 or under (middle), and for participants who were over 45 (right).

We observed that participants who were 45 or under (i.e., Group 1: $age <= 45$) found Tilt-Explore engaging, responsive, and easy-to-use, compared to those who were over 45 (i.e., Group 2: $age > 45$). This observation was also reflected in their NASA-TLX scores — Mean: 45.53 (SD: 17.13) for Group 1, and Mean: 66.66 (SD: 9.95) for Group 2. The difference of means was statistically significant, $F(1) = 7.360, p < .017$.

6.6.4 Personalized Search Strategy. Most participants devised their own search strategies with Tilt-Explore. For example, to find the target in task T1, some participants always started from the top-left corner by tilting the phone all the way to the top edge, and then to the left edge, until they reached the top-left corner. Then, they panned either horizontally or vertically using Tilt-Explore. If lost, they reoriented themselves by going to one of the corners. While panning, some participants missed the target (i.e., a book cover) multiple times even though the target was in the viewport. In addition, participants who required a higher magnification scale (8X or higher) tended to miss the target more frequently because they observed a fraction of it at a time. Most participants promptly responded to Tilt-Explore's announcements related to over-tilting–they promptly stopped Tilt-Explore by lifting their finger off the screen, reoriented their hand to a comfortable position, and put the finger back on the screen to resume panning. Finally, most participants commented on the transfer function that this function makes panning deterministic, similar to panning with the built-in touch gestures.

6.6.5 Association with Screen Readers. As mentioned earlier, all participants recognized the screen readers' Touch-Exploration feature in Tilt-Explore. They mentioned that they frequently used screen readers' functions, such as "Read-all" (in iOS) or "Read from next item" (in Android), to listen to the description of on-screen items from top to bottom, left to right. They were also familiar with basic screen reader gestures, such as a 1-finger swipe left/right to select the next/previous item; however, they used these gestures infrequently because they considered selecting an item using those gestures was slow. On the contrary, they did not experience the slowness with Tilt-Explore because they could quickly pan through the items, and Tilt-Explore selects an item automatically for them as the item glides under their fingertip.

7 DISCUSSION

Integration with Current Assistive Technologies. Tilt-Explore is built on top of existing assistive technologies for people with low vision. We envision Tilt-Explore to be a standalone mode in smartphones. Low-vision users can turn this mode on or off anytime, in a way similar to turning a screen magnifier or a screen reader on or off. In addition, Tilt-Explore enables screen readers' Touch-Exploration mode and allows users to issue built-in screen magnifier or screen reader gestures, such as 2- or 3-finger drags for panning, 2-finger pinch-to-zoom, 1-finger swipe left/right for selecting next/previous UI elements. Furthermore, accidentally triggering Tilt-Explore mode by touching the screen for long (e.g., 800ms) has a very low risk, if none, because tilt-to-pan does not register an action event (e.g., a tap event), which usually causes an unintended context switch. Moreover, the built-in long-press gestures are disabled in the Touch-Exploration mode, which Tilt-Explore activates by default. As such, when Tilt-Explore and a screen reader are both activated, users can still drag or swipe their one finger on the screen to select UI elments.

We learned that integrating new gestures in assistive technology is challenging. For example, as mentioned in Section 6.5.2, a seemingly unrelated gesture, such as "Split-Tap", sometimes conflicted with Tilt-Explore in accidentally issuing a tap event on a UI.

Single-Handed Interaction. Although Tilt-Explore is inherently a single-handed interaction, most participants used it with two hands. We offer several explanations in this regard. First, most low-vision users are accustomed to bi-manual interaction, like resting their phone on the palm of their non-dominant hand while issuing panning or zooming gestures with the dominant hand. Therefore, these users might carry over their current practice when using Tilt-Explore. Second, it is also possible that low-vision users might consider Tilt-Explore as an augmentation to built-in panning gestures and find switching between tilt-to-pan and 2- or 3-finger-drag-to-pan seamless. Third, many low-vision users, especially those who bring their phone close to their eyes, might find it convenient to control the tilt motion with their dominant hand when using Tilt-Explore. Thus, we have reasons to believe that most low-vision users would prefer a usable interaction more than using it with a single hand. We also believe that users would gradually learn to use Tilt-Explore single-handedly with time and practice.

Dexterity of Wrist Motion. Tilt-Explore can benefit low-vision users to pan a large space quickly—it is less error-prone and less frustrating. These benefits primarily come from the dexterity of wrist motion, harnessed by our transfer function and the Touch-Exploration mode. Unfortunately, the dependence on the wrist motion is also a limiting factor, as low-vision users often develop difficulty making this motion due to aging. To compensate for reduced wrist dexterity, we can consider pairing wrist motion with other modalities, such as back-of-device interaction [26], or augment it by muscle and tendon activities as done in *AssistiveTouch* in Apple Watches [39].

Reading Performance with Panning. As described in Section 6.5.4, reading magnified texts on smartphones via panning is inefficient (e.g., participants' reading speed was 28-31 WPM, compared to 200-250 WPM for people without vision impairments). This finding is consistent with prior work [33]. For example, Hallett et al. [33] reported that screen magnifiers cause discomfort because of their lack of support for word wrapping. Thus, designing a panning technique to facilitate faster reading on smartphones for low-vision users is an open problem.

Need for a Better Screen Magnification Algorithm. We realized that Tilt-Explore could not address all panning-related challenges, most of which stem from the fact that the screen magnifies scale all content uniformly, including whitespace (e.g., background) and non-whitespace (e.g., UI elements, texts). In this regard, Billah et al. [19] proposed a space reduction algorithm that scales whitespace and non-whitespace indiscriminately. We believe that such an algorithm can improve the efficiency of Tilt-Explore substantially because users need to pan less whitespace, which usually causes a loss of contextual information necessary for interacting with the content elements.

Future Work. We will explore the potential of Tilt-Explore for blind users who use screen readers exclusively on smartphones. For example, navigating a long list using screen readers' gestures (e.g., 1-finger swipe left or right) is slow and tedious. Tilt-Explore can benefit blind users in this scenario. We will further explore how Tilt-Explore can bridge the gap between screen magnifiers and screen readers to accommodate low-vision users who need to switch between these two assistive technologies manually.

8 CONCLUSION

In this paper, we proposed Tilt-Explore, a tilt-based single-handed panning gesture for low-vision users. We first identified the design space and design parameters from the literature for tilt-based interaction, refined the design parameters by conducting a study with 12 low-vision participants. The study shed insight into the usability challenges of existing screen magnifiers; the need for a tilt-to-pan gesture, or more broadly, an augmented panning gesture; and issues with existing tilt-based research in accommodating people with low vision. Informed by this study, we designed ***Tilt-Explore***, a usable tilt-to-pan mode for screen magnifiers that incorporates low-vision users' preferences and introduces features from screen readers. A second user study with 16 low-vision users suggests that Tilt-Explore is effective, less error-prone, offers a single-handed alternative, and augments the built-in panning gesture. Tilt-Explore fits in between a screen-reader and a screen-magnifier. In addition, low-vision users who are 45 or under are likely to prefer Tilt-Explore over the default. Similarly, low-users who need less magnification (e.g., scale < 4×) are likely to acquire visual targets faster with Tilt-Explore. The synergy of IMU sensors and touch gestures, as embodied in Tilt-Explore, can make interaction with smartphones more usable for people with low vision.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their insightful feedback. Research reported in this publication was supported in part by National Eye Institute (NEI) of the National Institutes of Health (NIH) under award number R01EY03008501A1 (subaward number 87527/2/1159967). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

[1] 2020. AccessibilityServiceInfo – Android Developers. Retrieved October 8, 2020 from https://developer.android.com/reference/android/accessibilityservice/AccessibilityServiceInfo#attr_android:canRequestTouchExplorationMode
[2] 2020. Android Magnification. https://support.google.com/accessibility/android/answer/6006949?hl=en
[3] 2020. Magnification – Android Accessibility Help. Retrieved October 8, 2020 from https://support.google.com/accessibility/android/answer/6006949?hl=en
[4] 2020. Speed Reading Facts. https://secure.execuread.com/facts/.
[5] 2020. Zoom in on the iPhone screen. Retrieved October 8, 2020 from https://support.apple.com/guide/iphone/zoom-iph3e2e367e/ios
[6] AFB. 2020. Glossary of Eye Conditions. http://www.afb.org/info/living-with-vision-loss/eye-conditions/12#L
[7] B. Agarwal and W. Stuerzlinger. 2013. WidgetLens: a system for adaptive content magnification of widgets. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*. British Computer Society, 2578052, 1–10.
[8] Android. 2018. Android Sensor Manager. https://developer.android.com/reference/android/hardware/SensorManager. (Accessed on 09/19/2018).
[9] Android. 2019. Android Sensors. https://developer.android.com/guide/topics/sensors/sensors_overview.html
[10] Apple Inc. 2020. Use AssistiveTouch on your iPhone, iPad, or iPod touch. https://support.apple.com/en-us/HT202658 [Online; accessed 01-Oct-2020].
[11] Jeff Avery, Sylvain Malacria, Mathieu Nancel, Géry Casiez, and Edward Lank. 2018. Introducing Transient Gestures to Improve Pan and Zoom on Touch Surfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 25, 8 pages. https://doi.org/10.1145/3173574.3173599
[12] Ravin Balakrishnan and I. Scott MacKenzie. 1997. Performance Differences in the Fingers, Wrist, and Forearm in Computer Input Control. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. Association for Computing Machinery, New York, NY, USA, 303–310. https://doi.org/10.1145/258549.258764
[13] Sandra Bardot, Anke Brock, Marcos Serrano, and Christophe Jouffrais. 2014. Quick-glance and In-depth Exploration of a Tabletop Map for Visually Impaired People. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine (IHM '14)*. ACM, New York, NY, USA, 165–170. https://doi.org/10.1145/2670444.2670465
[14] Joel F. Bartlett. 2000. Rock 'n' Scroll Is Here to Stay. *IEEE Comput. Graph. Appl.* 20, 3 (2000), 40–45. https://doi.org/10.1109/38.844371
[15] Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin Bederson, and Alex Zierlinger. Year. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. In *Proceedings of INTERACT*, Vol. 3. 57–64.
[16] Benjamin B. Bederson and James D. Hollan. 1994. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*. ACM, 192435, 17–26. https://doi.org/10.1145/192426.192435
[17] 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*. ACM, 2647357, 117–122. https://doi.org/10.1145/2642918.2647357
[18] Syed Masum Billah, Vikas Ashok, Donald E. Porter, and I.V. Ramakrishnan. 2017. Speed-Dial: A Surrogate Mouse for Non-Visual Web Browsing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 3132531, 110–119. https://doi.org/10.1145/3132525.3132531
[19] 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*. ACM, 3173594, 1–13. https://doi.org/10.1145/3173574.3173594
[20] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 985758, 519–526. https://doi.org/10.1145/985692.985758
[21] P. Blenkhorn and D. G. Evans. 2006. A Screen Magnifier Using "High Level" Implementation Techniques. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 4 (2006), 501–504. https://doi.org/10.1109/TNSRE.2006.886728
[22] P. Blenkhorn, D. G. Evans, and A. Baude. 2002. Full-screen magnification for windows using DirectX overlays. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 10, 4 (2002), 225–231. https://doi.org/10.1109/TNSRE.2002.806835
[23] John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189 (1996), 194.

[24] Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Amaury Trujillo. 2015. Exploring Visually Impaired People's Gesture Preferences for Smartphones. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*. ACM, 2808448, 94–101. https://doi.org/10.1145/2808435.2808448

[25] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1499–1508. https://doi.org/10.1145/2702123.2702425

[26] Wenzhe Cui, Suwen Zhu, Zhi Li, Zheer Xu, Xing-Dong Yang, Iv Ramakrishnan, and Xiaojun Bi. 2021. BackSwipe: Back-of-device Word-Gesture Interaction on Smartphones. 1–12. https://doi.org/10.1145/3411764.3445081

[27] Julie Fraser and Carl Gutwin. 2000. A framework of assistive pointers for low vision users. In *Proceedings of the fourth international ACM conference on Assistive technologies*. ACM, 354329, 9–16. https://doi.org/10.1145/354324.354329

[28] George W. Furnas and Benjamin B. Bederson. 1995. Space-Scale Diagrams: Understanding Multiscale Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 234–241. https://doi.org/10.1145/223904.223934

[29] E. Grandjean and H. Oldroyd. 1980. *Fitting the Task to the Man: An Ergonomic Approach*. Taylor & Francis. https://books.google.com/books?id=CFtRAAAAMAAJ

[30] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1055012, 281–290. https://doi.org/10.1145/1054972.1055012

[31] Yves Guiard. 2013. Asymmetric Division of Labor in Human Skilled Bimanual Action. *Journal of Motor Behavior* 19 (08 2013), 486–517. https://doi.org/10.1080/00222895.1987.10735426

[32] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. 2004. Object pointing: a complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 1006060, 9–16.

[33] Elyse C. Hallett, Wayne Dick, Tom Jewett, and Kim-Phuong L. Vu. 2018. *How Screen Magnification with and Without Word-Wrapping Affects the User Experience of Adults with Low Vision*. Springer International Publishing, Cham, 665–674. https://doi.org/10.1007/978-3-319-60492-3_63

[34] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. 1998. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., 274647, 17–24. https://doi.org/10.1145/274644.274647

[35] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. 2000. Sensing Techniques for Mobile Interaction. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 91–100. https://doi.org/10.1145/354401.354417

[36] Ken Hinckley and Hyunyoung Song. 2011. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 801–810. https://doi.org/10.1145/1978942.1979059

[37] Juan Pablo Hourcade, Christopher M. Nguyen, Keith B. Perry, and Natalie L. Denburg. 2010. Pointassist for older adults: analyzing sub-movement characteristics to aid in pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1753494, 1115–1124. https://doi.org/10.1145/1753326.1753494

[38] https://developer.apple.com/. 2020. Adaptivity and Layout - Visual Design - iOS - Human Interface Guidelines - Apple Developer. https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/.

[39] Apple Inc. 2021. Apple Previews Powerful Software Updates Designed for People with Disabilities. https://www.apple.com/newsroom/2021/05/apple-previews-powerful-software-updates-designed-for-people-with-disabilities/

[40] Paul Kabbash and William A. S. Buxton. 1995. The "prince" technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., 223939, 273–279. https://doi.org/10.1145/223904.223939

[41] Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 1414487, 73–80. https://doi.org/10.1145/1414471.1414487

[42] Shaun K. Kane, Jacob O. Wobbrock, and Richard E. Ladner. 2011. Usable gestures for blind people: understanding preference and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1979001, 413–422. https://doi.org/10.1145/1978942.1979001

[43] 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*. ACM Press/Addison-Wesley Publishing Co., 223919, 114–121. https://doi.org/10.1145/223904.223919

[44] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.

[45] I. Scott MacKenzie and Robert J. Teather. 2012. FittsTilt: The Application of Fitts' Law to Tilt-Based Interaction. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (NordiCHI '12)*. Association for Computing Machinery, New York, NY, USA, 568–577. https://doi.org/10.1145/2399016.2399103

[46] Material Design. 2020. Understanding layout - Material Design. https://material.io/design/layout/understanding-layout.html.

[47] Stephania Medryk and I Scott MacKenzie. 2013. A comparison of accelerometer and touch-based input for mobile gaming. In *International Conference on Multimedia and Human-Computer Interaction-MHCI 2013*. 117–1.

[48] Alexander Ng, Stephen A. Brewster, and John H. Williamson. 2014. Investigating the Effects of Encumbrance on One- and Two- Handed Interactions with Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1981–1990. https://doi.org/10.1145/2556288.2557312

[49] I. Oakley and S. O'Modhrain. 2005. Tilt to scroll: evaluating a motion based vibrotactile mobile interface. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. 40–49. https://doi.org/10.1109/WHC.2005.138

[50] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-Supported Text Entry for Very Small Devices. *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*, 201–204. https://doi.org/10.1145/571985.572013

[51] Martin Pielot, Benjamin Poppinga, Wilko Heuten, and Susanne Boll. 2011. A tactile compass for eyes-free pedestrian navigation. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*. Springer-Verlag, 2042179, 640–656.

[52] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. 2009. Tilt Techniques: Investigating the Dexterity of Wrist-based Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1943–1952. https://doi.org/10.1145/1518701.1518997

[53] Jun Rekimoto. 1996. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM, 237115, 167–168. https://doi.org/10.1145/237091.237115

[54] RotoView. [n. d.]. Smart Scroll Technology for Smartphones and Other Handheld Devices. http://rotoview.com/index.htm

[55] Jaime Ruiz and Yang Li. 2011. DoubleFlip: a motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1979341, 2717–2720. https://doi.org/10.1145/1978942.1979341

[56] Jaime Ruiz, Yang Li, and Edward Lank. 2011. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1978971, 197–206. https://doi.org/10.1145/1978942.1978971

[57] Sonja Rümelin, Enrico Rukzio, and Robert Hardy. 2011. NaviRadar: a novel tactile information display for pedestrian navigation. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2047234, 293–302. https://doi.org/10.1145/2047196.2047234

[58] Samsung. [n. d.]. How do I use Motion and Gestures to control my Samsung Galaxy S4? https://www.samsung.com/us/support/answer/ANS00043991/

[59] David Small and Hiroshi Ishii. 1997. Design of spatially aware graspable displays. In *CHI '97 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1120437, 367–368. https://doi.org/10.1145/1120212.1120437

[60] 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 (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 361–362. https://doi.org/10.1145/3132525.3134812

[61] 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*. ACM, 2982168, 171–180. https://doi.org/10.1145/2982142.2982168

[62] Robert J. Teather and I. Scott MacKenzie. 2014. Position vs. Velocity Control for Tilt-based Interaction. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 51–58. http://dl.acm.org/citation.cfm?id=2619648.2619658

[63] Robert J. Teather, Andrew Roth, and I. Scott MacKenzie. 2017. Tilt-Touch synergy: Input control for "dual-analog" style mobile games. *Entertainment Computing* 21 (2017), 33–43. https://doi.org/10.1016/j.entcom.2017.04.005

[64] Mary Frances Theofanos and Janice Redish. 2005. Helping Low-vision and Other Users with Web Sites That Meet Their Needs: Is One Site for All Feasible? *Technical Communication* 52, 1 (2005), 9–20. http://www.ingentaconnect.com/content/stc/tc/2005/00000052/00000001/art00002

[65] Theophanis Tsandilas, Caroline Appert, Anastasia Bezerianos, and David Bonnet. 2014. Coordination of Tilt and Touch in One- and Two-handed Use. In *Proceedings*

of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 2001–2004. https://doi.org/10.1145/2556288.2557088

[66] Gregg C. Vanderheiden. 1996. Use of Audio-Haptic Interface Techniques to Allow Nonvisual Access to Touchscreen Appliances. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 40, 24 (1996), 1266–1266. https://doi.org/10.1177/154193129604002430

[67] Lars Weberg, Torbjörn Brange, and Åsa Wendelbo Hansson. 2001. A Piece of Butter on the PDA Display. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. Association for Computing Machinery, New York, NY, USA, 435–436. https://doi.org/10.1145/634067.634320

[68] Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 81–90. https://doi.org/10.1145/964696.964705

[69] Won Kim, F. Tendick, S. Ellis, and L. Stark. 1987. A comparison of position and rate control for telemanipulations with consideration of manipulator system dynamics. *IEEE Journal on Robotics and Automation* 3, 5 (Oct. 1987), 426–436. https://doi.org/10.1109/JRA.1987.1087117

[70] Aileen Worden, Nef Walker, Krishna Bharat, and Scott Hudson. 1997. Making computers easier for older adults to use: area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 258724, 266–271. https://doi.org/10.1145/258549.258724

[71] Hui-Shyong Yeo, Xiao-Shen Phang, Steven J. Castellucci, Per Ola Kristensson, and Aaron Quigley. 2017. Investigating Tilt-based Gesture Keyboard Entry for Single-Handed Text Entry on Large Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3025520, 4194–4202. https://doi.org/10.1145/3025453.3025520

[72] Yuhang Zhao, Sarit Szpiro, Lei Shi, and Shiri Azenkot. 2019. Designing and Evaluating a Customizable Head-Mounted Vision Enhancement System for People with Low Vision. *ACM Trans. Access. Comput.* 12, 4, Article 15 (Dec. 2019), 46 pages. https://doi.org/10.1145/3361866

[73] Zhengxuan Zhao, Pei-Luen Patrick Rau, Ting Zhang, and Gavriel Salvendy. 2009. Visual search-based design and evaluation of screen magnifiers for older and visually impaired users. *Int. J. Hum.-Comput. Stud.* 67, 8 (2009), 663–675. https://doi.org/10.1016/j.ijhcs.2009.03.006